01-30-08 IFW

OIPE IAP84 JAN 29 2008 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Christopher Vienneau	Examiner:	Thuy Thanh Osberg
Serial No.:	10/619,758	Group Art Unit:	2179
Filed:	July 15, 2003	Docket:	G&C 30566.335-US-01
Title:	PROCESSING IMAGE DATA		

CERTIFICATE OF MAILING UNDER 37 CFR 1.10
'Express Mail' mailing label number: EV 887 247 188 US
Date of Deposit: January 29, 2008
I hereby certify that this paper or fee is being deposited with the United States Postal Service 'Express Mail Post Office To Addressee' service under 37 CFR 1.10 and is addressed to: , P.O. Box 1450, Alexandria, VA 22313-1450.

By: [signature]
Name: Kathleen Krochko

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

We are transmitting herewith the attached:

☒ Transmittal sheet, in duplicate, containing a Certificate of Mailing under 37 CFR 1.10.
☒ Certified copy of United Kingdom application No. GB 02 16844.1, filed July 19, 2002, the right of priority of which is claimed under 35 U.S.C. 119.
☒ Return postcard.

Please consider this a PETITION FOR EXTENSION OF TIME for a sufficient number of months to enter these papers, if appropriate.

Please charge all fees to Deposit Account No. 50-0494 of Gates & Cooper LLP. A duplicate of this paper is enclosed.

Customer Number 55895
GATES & COOPER LLP
Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, CA 90045
(310) 641-8797

By: [signature]
Name: Jason S. Feldmar
Reg. No.: 39,187
JSF/kmk

(PTO TRANSMITTAL - GENERAL)

Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) [illegible]f the Deregulation and Contracting Out Act 1994, to sign and issue certificates on behalf of [illegible]he Comptroller-General, hereby certify that annexed hereto is a true copy of the documents [illegible]s originally filed in connection with patent application GB0216844.1 filed on 19 July 2002, [illegible]gether with the Statement of inventorship and of right to grant of a Patent (Form 7), which [illegible]s subsequently filed.

[illegible]o certify that the attached copy of the request for grant of a Patent (Form 1) bears an a[illegible]ndment, effected by this office, following a request by the applicant and agreed to by the C[illegible]ptroller-General.

In [illegible]rdance with the Patents (Companies Re-registration) Rules 1982, if a company named in t[illegible]ertificate and any accompanying documents has re-registered under the Companies Act [illegible]0 with the same name as that with which it was registered immediately before re-regis[illegible]on save for the substitution as, or inclusion as, the last part of the name of the words "publ[illegible]mited company" or their equivalents in Welsh, references to the name of the compa[illegible]in this certificate and any accompanying documents shall be treated as references to the na[illegible]with which it is so re-registered.

[illegible]n accor[illegible]e with the rules, the words "public limited company" may be replaced by p.l.c., [illegible]lc, P.L.[illegible] PLC.

[illegible]e-registra[illegible] under the Companies Act does not constitute a new legal entity but merely [illegible]bjects the[illegible]pany to certain additional company law rules.

Signed [signature]

Dated 18 January 2008

THE PATENT OFFICE
A
19 JUL 2002
RECEIVED BY FAX

19JUL02 E734742-1 003028
P01/7700 0.00-0216844.1

Patents Form 1/77

Patents Act 1977

The Patent Office

1/77

Request for grant of a patent
Grant

The Patent Office
Concept House
Cardiff Road
Newport
Gwent NP10 8QQ

1.	Your reference	2034-P578-GB
2.	Patent application number 19 JUL 2002 0216844.1	New
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	AUTODESK CANADA INC 10 Duke Street Montreal Quebec Canada H3C 2L7
	08378069001 Patents ADP number *(if you know it)*	
	If the applicant is a corporate body, give the country/state of its incorporation	Quebec, Canada
4.	Title of the invention	Editing Image Data
5.	Name of your agent	~~ATKINSON BURRINGTON~~
	"Address for service" in the United Kingdom to which all correspondence should be sent MARKS & Clerk 57-60 Lincoln's Inn Fields LONDON WC2A 3LS	~~25-29 President Buildings President Way Sheffield S4 7UR GB~~
	Telephone No:	0114 275 2400
	Patents ADP number	7807043001

		Country	Priority application number *(if you know it)*	Date of filing *(day/month/year)*
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	N/A	N/A	N/A

		Number of earlier application	Date of filing (day/month/year)
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	N/A	N/A

8.	Is a statement of inventorship and of right to grant of a patent required in support of this request?	Yes

0042197 19-Jul-02 04:25

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	37
Claim(s)	05
Abstract	01
Drawings	20 only

DML

10. If you are also filing any of the following, state how many against each item.

Priority documents	N/A
Translations of priority documents	N/A
Statement of inventorship and right to grant of a patent (*Patents Form 7/77*)	None
Request for preliminary examination and search (*Patents Form 9/77*)	One
Request for substantive examination (*Patents Form 10/77*)	None
Any other documents (*Please specify*)	

11. I/We request the grant of a patent on the basis of this application.

Signature

Date Friday, 19 July 2002

12. Name and daytime telephone number of person to contact in the United Kingdom

RALPH ATKINSON CPA
0114 275 2400

LA-07-150

0042197 19-Jul-02 04:25

Patents Form 7/77

Patents Act 1977

7/77

The Patent Office
Concept House
Cardiff Road
Newport
Gwent NP10 8QQ

Statement of inventorship and of right to grant of a patent

1.	Your reference	2034-P578-GB
2.	Patent application number	02 16 844.1
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	AUTODESK CANADA INC 10 Duke Street Montréal Quebec Canada H3C 2L7
4.	Title of the invention	EDITING IMAGE DATA
5.	State how the applicant*(s)* derived the right from the inventor*(s)* to be granted a patent	The applicant derived the right to the invention by virtue of a contract of employment
6.	How many, if any, additional Patents Forms 7/77 are attached to this form? *(see note (c))*	None

7\. I/We believe that the person*(s)* named over the page *(and on any extra copies of this form)* is/are the inventor*(s)* of the invention which the above patent application relates to.

Signature

Date Thursday, 22 August 2002

8.	Name and daytime telephone number of person to contact in the United Kingdom	RALPH ATKINSON CPA 0114 275 2400

Patents Form 7/77

Enter the full names, address and postcodes of the inventors in the boxes and underline the surnames

Chris VIENNEAU
350 Prince Arthur
Apt. D1208
Montréal
Quebec
Canada H2X 3R4

828407700l

Patents ADP number:

Juan Pablo DI LELLE
334 Terrasse St Denise #304
Montréal
Quebec
Canada H2X 1E8

Patents ADP number: 8275570001

Patents ADP number:

Patents ADP number

SR-08-135

DUPLICATE

Editing Image Data

Background of the Invention

1. Field of the Invention

The present invention relates to a method and apparatus for processing image data, wherein components of said image data are defined by structured data processing nodes, which require editing.

2. Description of the Related Art

Many post production processes have been devised and employed over the years to enhance movie productions or video films with what is commonly referred to as "special effects". Such image enhancement have long been provided by means of using dedicated hardware, either as a scale model to be filmed and subsequently composited in post production or, more recently, by means of effects generators such as computer apparatus configured to output rendered image components to be also subsequently composited.

Technical advances in image processing systems have enabled the generalisation of the "blue screen" technique in video environments and "green screen" technique in cinematographic environments, whereby actors are filmed in a studio configured with blue or green saturated surroundings in order to generate a clip of foreground image frames. An alternative clip of background image frames is subsequently generated and a compositing process allows an editor to seamlessly blend the foreground and background image frames by means of keying part of the corresponding video signals, for instance the luminance or chrominance signal.

A problem inherent to the above image processing techniques was that effects generation could not take place in real-time, that is all of the various components in each image frame within a clip of such image frames would have to be shot in the case of real actors and/or situations committed to cinematographic film, or rendered in the case of computer graphics images, prior to the compositing process. Thus, if the appearance of any of such image components was artistically unsatisfactory, the only possible manner of remedying this problem would be to do another shot on new cinematagraphic film or amend parameters in the computer to generate a new version of the computer-rendered image components.

Modern image processing systems overcome the above problem by means of providing real-time image data processing capability, whereby every image component within an image frame exists as a hierarchical sub-structure of data processing nodes within a main structure, which thus defines the entire image frame. An image editor using such a modem image processing system can amend parameters and/or data in any of said data processing nodes to aesthetically improve any image component within an image frame and assess the effectiveness of his editing in real-time.

An important problem has however developed within modem image processing systems as described above, as the size of modern movie image frames or high definition video image frames has dramatically increased both in resolution and in the number of components therein, to such an extent that it has now become impractical to display one such image frame at full resolution, its corresponding structure of data processing nodes and further user-operable controls to edit the data and/or

parameters of said nodes in said image processing systems. Indeed, conventional display devices of image processing systems can barely accommodate such image frames at full resolution, often as not exceeding 2000 pixels by 2000 pixels. Moreover, a structure of data processing nodes corresponding to any such image frame often as not includes hundreds or even thousands of such nodes to the extent that the readability of such a structure becomes problematic from a user's point of view when displayed as an alternative to said image frame.

A need therefore exists for a method and apparatus of processing image data including a plurality of components defined by a plurality of data processing nodes, which is more intuitive to a user and allows said user to rapidly identify a relevant edit data processing node and edit the data and/or parameters therein.

Brief Summary of the Invention

According to the present invention, a method of processing image data is provided which comprises storage means, processing means and display means, wherein said image data includes a plurality of components defined by a plurality of hierarchical data processing nodes, said storage means are configured to store said image data and said processing means are configured to perform the steps of

generating first image data by means of processing said hierarchical data processing nodes;

outputting said first image data to said display means;

restructuring said hierarchy in response to user input data; and

generating second image data by means of processing said restructured hierarchical editing nodes; and
outputting said second image data to said display means.

Brief Description of the Several Views of the Drawings

Figure 1 shows an image processing system operated by an editor;

Figure 2 details the hardware components of the image processing system of *Figure 1* in further detail;

Figure 3 details the operational steps according to which a user operates the image processing system of *Figures 1* and *2*;

Figure 4 shows the contents of the main memory shown in *Figure 2* subsequently to the instructions processing start shown in *Figure 3*;

Figure 5 details the loading and processing of image data as described in *Figure 4*;

Figure 6 shows the video display unit shown in *Figure 1* as displaying an image frame within the graphical user interface (GUI) of the application shown in *Figures 4* and *5*;

Figure 7 shows a process tree as an example of the main structure shown in *Figures 4* and *5*;

Figure 8 provides a graphical representation of the data generated at each data processing node of the process tree shown in *Figure 7* when said processing nodes are processed;

Figure 9 further details the main structure shown in *Figures 4, 7* and *8* in terms of hierarchically structured data processing nodes;

Figure 10 shows the process tree shown in *Figures 7, 8* and *9* within the GUI shown in *Figure 6* according to the prior art;

Figure 11 details the operational steps according to which relevant image data is identified and edited according to the present invention;

Figure 11b details the operational steps according to which sub-structures of data processing nodes are identified;

Figure 11c details the operational steps according to which an image component is selected by means of its corresponding sub-structure;

Figure 11d details the operational steps according to which compatible data processing nodes are identified and selected in a different sub-structure;

Figure 12 shows the image frame within the graphical user interface (GUI) shown in *Figure 6* according to the present invention;

Figure 13 shows the process tree shown in *Figures 7, 8* and *9* within the GUI shown in *Figure 6* according to an alternative embodiment of the present invention;

Figure 14 shows the image frame within the graphical user interface (GUI) shown in *Figure 12,* wherein navigation input data shown in *Figure 11* has been provided;

Figure 15 shows the process tree shown in *Figure 13* according to an alternative embodiment of the present invention, wherein navigation input data shown in *Figure 11* has been provided;

Figure 16 shows the image frame within the graphical user interface (GUI) shown in *Figures 12* and *14,* wherein alternative navigation input data shown in *Figure 11* has been provided;

Figure 17 shows the process tree shown in *Figures 13* and *15* according to an alternative embodiment of the present invention, wherein alternative navigation input data shown in *Figure 11* has been provided;

Written Description of the Best Mode for Carrying Out the Invention

Figure 1

An image processing system such as a post-production station is illustrated in *Figure 1*. A processing system **102**, such as an Octane™ produced by Silicon Graphics Inc., supplies image signals to a video display unit **103**. Moving image data is stored on a redundant array of inexpensive discs (RAID) **104**. The RAID is configured in such a way as to store a large volume of data, and to supply this data at a high bandwidth, when required, to the processing system **102**. The operator controls the processing environment formed by the processing system **102**, the video monitor **103** and the RAID **104**, by means of a keyboard **105**, and a stylus-operated graphics tablet or a mouse **106**. The processing system shown *in Figure 1* is optimal for the purpose of processing image and other high bandwidth data. In such a system, the instructions for controlling the processing system are complex. The invention relates to any computer system where processing instructions are of significant complexity.

Instructions controlling the processing system **102** may be installed from a physical medium such as a CDROM or DVD disk **107**, or over a network **108**, including the Internet. These instructions enable the processing system **102** to interpret user commands from the keyboard **105** and the graphics tablet **106**, such that image data, and other data, may be viewed, edited and processed.

Figure 2

The processing system **102** shown in *Figure 1* is detailed in *Figure 2*. The processing system comprises two central processing units **201** and **202** operating in parallel. Each of these processors is a MIPS R11000 manufactured by MIPS Technologies Incorporated, of Mountain View, California. Each of these processors **201** and **202** has a dedicated secondary cache memory **203** and **204** that facilitate per-CPU storage of frequently used instructions and data. Each CPU **201** and **202** further includes separate primary instruction and data cache memory circuits on the same chip, thereby facilitating a further level of processing improvement. A memory controller **205** provides a common connection between the processors **201** and **202** and a main memory **206**. The main memory **206** comprises two gigabytes of dynamic RAM.

The memory controller **205** further facilitates connectivity between the aforementioned components of the processing system **102** and a high bandwidth non-blocking crossbar switch **207**. The switch makes it possible to provide a direct high capacity connection between any of several attached circuits. These include a graphics card **208**. The graphics card **208** generally receives instructions from the processors **201** and **202** to perform various types of graphical image rendering processes, resulting in images, clips and scenes being rendered in real time on the monitor **102**. A high bandwidth SCSI bridge **209** provides an interface to the RAID **104**, and also, optionally, to a digital tape device, for use as backup.

A second SCSI bridge **210** facilitates connection between the crossbar switch **207** and a DVD/CDROM drive **211**. The DVD drive provides a convenient way of receiving large quantities of instructions and

data, and is typically used to install instructions for the processing system **101** onto a hard disk drive **212**. Once installed, instructions located on the hard disk drive **212** may be fetched into main memory **206** and then executed by the processors **201** and **202**. An input output (I/O) bridge **213** provides an interface for the graphics tablet **106** and the keyboard **105**, through which the user is able to provide instructions to the processing system **102**.

Figure 3

Terminal **102** is switched on by user **101** at step **301**, such that central processing unit instructions may be permanently loaded onto hard disk drive **212** or temporarily loaded within main memory **206** from CD ROM or DVD ROM **107**, network server **109** or the Internet **108**.

Upon completing the loading operation of step **302**, the application starts at step **303**, whereby the instructions thereof are processed by central processing units **201** and **202**. At step **304**, image data from a single frame or, alternatively, from a clip of frames is acquired from frame store **104** such that it can be displayed to user **101** on video display unit **103** for subsequent editing at step **305**.

Upon observing the frame displayed at step **304**, user **101** is thus at liberty to modify any or all of the various components of the image data at step **305**. The final edited image data may eventually be stored at frame store **104** upon completing the required level of image data editing.

At step **307**, a question is asked as to whether another image frame or another clip of image frames require processing by image processing system **102**, whereby control is returned to step **304** such that a new frame

or clip of frames can be acquired from frame store **104** if answered in the affirmative. Alternatively, if the question asked at step **307** is answered in the negative then user **101** is at liberty to eventually switch the image processing system **102** off at step **308**.

The contents of main memory **206** subsequent to the instructions processing start of step **303** and image data acquisition of step **304** are further detailed in *Figure 4*.

Figure 4

An operating system is shown at **401** which comprises a reduced set of instructions for CPUs **201** and **202**, the purpose of which is to provide image processing system **102** with basic functionality. Examples of basic functions include access to files stored on hard disk drive **212** or DVD/CD ROM drive **211** and management thereof, network connectivity with frame store **104**, server **109** and the Internet **108**, interpretation and processing of the input from keyboard **105** and graphic tablet or mouse **106** and graphical data or binary data output. In the example, the operating system is IRIX™ provided by Silicon Graphics Inc, but it will be apparent to those skilled in the art that the instructions according to the present invention may be easily adapted to function under different other known operating systems, such as Windows™ provided by the Microsoft Corporation of Redmond, California or LINUX which is freely distributed.

An application is shown at **402** which comprises the instructions loaded at step **302** that enable the image processing system **102** to perform steps **303 to 307** according to the invention within a specific graphical user interface displayed on VDU **103**.

Application data is shown at **403** and comprises various sets of user input dependent data and user input independent data according to which the application shown at **402** processes image data. Said application data primarily includes main structure data **404**, which references the entire history of the image data as loaded at step **304** and comprehensively defines each component within an image frame in terms of hierarchically-structured data processing nodes, an example of which will be described further below. Accordingly, application data **403** also includes data **405** defining the various types of data processing nodes present within the structure or which may be inserted therein as a consequence of image data editing.

Further to the main structure data **404** and nodes definition data **405**, application data **403** includes node data **406** to **411** to be processed in order to generate the current image frame, i.e. the parameters and data which, when processed by their respective data processing nodes, generate the various components of said image frame.

In the example, node data comprises three-dimensional models **406** defined as a plurality of polygons or possibly non-uniform rational B-splines (NURBS). Node data also comprises bitmap files **407** to be applied as textures to said three-dimensional models **406** lightmaps 408 wholly or partially.

Node data also comprises three-dimensional positional data **409**, possibly in the form of vectors, to define scaling and tracking of said three-dimensional models **406** within a three-dimensional space. Node data also comprises RGB data **410** defining an image frame derived from film and digitally stored in frame store **104**. Node data eventually comprises sound

files **411**, for instance the portion of clip soundtrack corresponding to the image frame being edited. It will be easily understood by those skilled in the art that the above data types are for illustrative purposes only and the list described is non-exhaustive. Said data types relate to the type of data processing nodes required to define and generate the image frame components, a potentially infinite combination of which there may exist.

Frame data is shown at **412**, which comprises user input independent data defining image frames acquired from frame store **104**. Said definition includes a ClipID **413** referencing which clip a frame to be edited is part of and a FrameID **414** referencing which frame is to be edited within said clip. Frame data **412** also includes frame resolution **415** indicating the frame size in terms of picture screen elements, known to those skilled in the art as pixels, such that application **402** may appropriately configure output data for display at full resolution.

Finally, user input data is shown at **416**, which comprises user input dependent data identifying parameters and/or data input by user **101** by means of keyboard **105** and graphic tablet or mouse **106** for the purpose of editing the above image data.

Figure 5

The loading and processing of image data as described in *Figure 4* at step **304** is further detailed in *Figure 5*.

Initially, the main structure **404** is loaded at step **501**, whereby all of the data processing nodes are initiated, along with node type data **405** and node data **406** to **411**. At step **502**, said data processing nodes are processed according to the hierarchy defined by said main structure,

whereby all of the data processing nodes process their respective node data **406** to **411**. The totality of image components defined by said main structure **404** are thus output to VDU display **103** as an image frame for user **101** to subsequently edit at step **503**.

A question is asked at step **504** as to whether a new image component is required. User **101** may wish to impart some creative input to the image frame as it is currently defined, for instance by means of new visual effects or further objects within the scene defined by said image frame. If the question is answered in the affirmative, a data processing node is initiated at step **505**.

In the preferred embodiment of the present invention, said node may either already have been initiated according to step **501** or may be a new type of application-compatible node loaded ed from CD ROM or DVD ROM 107 or network server **109** or downloaded from the Internet **108**. Upon completing the initiation step **505**, the new data processing node is registered within the main structure **404** at step **506**, in terms of its dependency with regard to all of the other data processing nodes already referenced therein, a more detailed description of which will be provided further below. Alternatively, the question of step **504** is answered negatively, whereby image data can now be edited according to step **305**.

Figure 6

The image frame generated according to step **502** for display according to step **503** is preferably output to the video display unit **103** of image processing system **102**, within a Graphical User Interface (GUI), which is shown in *Figure 6*.

The GUI **601** of application **502** is divided into a plurality of functional areas, portions of which are user-operable. A first area **602** displays image data **603** acquired at step **304**. A second area **604** comprises user-operable conventional clip navigation widgets allowing user **101** to rewind **605**, backward play **606**, pause **607**, stop **608**, forward play **609** or fast-forward **610** the sequential order of image frames within a clip if user **101** acquired a clip at step **304**.

A counter area **611** is provided in close proximity to area **602** and divided into an hour counter **612**, minute counter **613**, seconds counter **614** and frame counter **615**, whereby said frame counter may operate in base twenty-four, base thirty or base sixty depending upon the provenance of the clip, eg respectively cinema, video or high definition TV. Said counter area **611** enables user **101** to accurately determine where the currently displayed image frame is located within the complete sequence of the clip.

A user-operable switch **616** is also provided within GUI **601**, the manipulation of which by user **101** via preferably but not exclusively, mouse **106** allows GUI **601** to alternatively display the image frame **603** or a graphical representation of the corresponding main structure **404** defining the components thereof.

A user-operable conventional bar **617** of menu commands is provided in the left uppermost area of GUI **601**, which includes a plurality of shortcuts to facilitate frame or file access, application configuring and other such conventional application functions. A user-operable conventional bar **618** of GUI sizing or application termination icons are provided in the right uppermost corner of **601**.

In the example, user **101** acquires image data **403** and **412** defining image frame **603** from frame store **104** according to step **304**, whereby the components thereof are displayed in display area **602** of GUI **601** according to step **503**.

-

Figure 7

A simplified example of a main structure **404** defining the components of image frame **603**, also known to those skilled in the art as an edit decision list or process tree, is shown in *Figure 7*.

A process tree consists of hierarchical, sequentially-linked data processing nodes, each of which specifies a particular processing task required in order to eventually achieve scene output data **701**, for instance under the form of frame **603**. Traditionally, an output sequence **701** will comprise both image data and audio data. Accordingly, the scene will thus require the output from an image-keying node **702** and the output of a sound mixing node **703**. The image-keying node **702** calls on a plurality of further graphic data processing nodes to obtain all of the input data it requires to generate the desired image components. In effect, all of the nodes in the process tree define 'branches' of parent and children nodes and sub-divisions thereof and, insofar as the graphical nodes of the tree shown in *Figure 7* are concerned, each branch of nodes born from the ultimate graphical parent node **702** defines a layer. The purpose of image-keying node **702** is thus to composite the layers, e.g. superimpose the four layers shown in the example, which are further detailed below.

In the example, the desired output image frame includes a three-dimensional computer generated object composited with a background also

consisting of a plurality of three-dimensional objects superimposed over a background texture.

The image-keying node **702** thus initially requires a frame as RGB data **410** from frame node **704**, which is subsequently processed by a colour-correction processing node **705** and subjected to positional data **709** of a motion tracking processing node **706**. The composited three-dimensional model **406** generated by three-dimensional modelling node **707**, to which is applied a bitmap file **407** as a texture by the texturing node **708** and appropriate lightmap **408** by artificial light processing node **709**, is scaled by scaling node **710** and also subjected to positional data **709** of a motion tracking processing node **711**, such that it is seamlessly composited within the colour corrected frame **704**.

In so far as the background is concerned, the image keying processing node **702** also requires a uniform texture **407** from a texturing node **712**, the functionality of which is similar to the texturing node **708**, to which is applied the colour-correction of a colour-correction processing node **713**, the functionality of which is similar to node **705**. The image-keying processing node **702** finally requires to overlay the plurality of simple three-dimensional models **406** generated from the three-dimensional modelling node **714**, which are appropriately lit with lightmaps **408** by the artificial light processing node **715** and motion-tracked with positional data **409** by means of the motion-tracking processing node **716** over the colour corrected-texture **711** before overlaying the composited frame of node **704** on top of the composited background. The scene **701** is completed by associating the output of sound mixing node **703** with the output of image keying node **702**.

Figure 8

It is common for process trees such as detailed in *Figure 7* to incorporate hundreds and potentially even thousands of logically-linked data processing nodes configured as parent processing nodes and children processing nodes, each of which symbolises a functionality applied to some form of data or other.

The respective output data of each parent and children nodes **701** to **716** of the process tree detailed in *Figure 7* are graphically shown in *Figure 8* in order to illustrate the application data **403** processed at step **502**.

The generation of all of the image components requires processing of all of the data processing nodes within the process tree. Said processing begins at the leftmost last child node **704**, whereby an operational function of said child node **704** is invoked in order to fetch a frame **801**, which depicts a real plane photographed in front of a blue or green background in order to facilitate subsequent keying processes, depending upon whether the image was respectively shot for video or cinematography.

Node **705** is a parent node of node **704** and subsequently pulls the frame **801** from node **704**, and its colour correction operational function modifies the colour **802** of said frame by applying a processing rule to the RGB values of every pixel in said frame. According to the invention, it is known to parent node **705** that frame **801** comprises a finite number of pixels corresponding to the resolution of frame **801** as the definition **412** of the frame is an array of pixels and its resolution **415** is for instance the number of pixels in a 24D high-definition cinematographic frame. The

parent node **706** of node **705** subsequently pulls the data of said node and calls an operational function designed to orient the colour-corrected frame within a three dimensional volume **803**.

The next branch, or layer, depending from the parent node **702** is followed and thus proceeds to node **707**, whereby a three dimensional computer generated model **406** of a plane is generated as component **804** by operational functions of said node **707**. Its parent node **708** subsequently pulls said three-dimensional computer-generated model 804 in order to apply a "steel" bitmap texture 805 to each polygon of said three-dimensional model. It is known to node **708** that the three-dimensional model is composed of polygons defined by tessellating the smallest component of the model, which are vertices.

Processing node **709** subsequently applies an artificial lighting algorithm **408** at **806** to the textured three-dimensional model and processing node **710** can subsequently scale the lit (**806**), textured (**805**) three-dimensional model **804** at **807**. The parent node **711** of node **710** subsequently pulls the data of said node and calls an operational function designed to animate the the composited plane **807** within a three dimensional volume **808**, known to those skilled in the art as motion-tracking.

With respect to the topmost graphical parent node **702** within the process tree, two further branches respectively defined by nodes **712**, **713** and **714** to **716** have to be processed before it pulls the input data and processes said data itself. A "sky" bitmap texture **809** is thus generated by node **712** which is subsequently colour-corrected at parent node **713** using the same operational function as was invoked by colour-correction

processing node **705** to process the frame **801**.

Similarly, a computer-generated three-dimensional "clouds" model **811** is generated by node **714** utilising potentially the same operational function as was invoked by node **707** to generate the "plane" three dimensional model. The three-dimensional model **811** is subsequently lit (**812**) at parent node **715** using potentially the same lighting algorithm of the operational function called at node **709**. The lit (**812**) three-dimensional model **810** is subsequently motion-tracked (**813**) at processing node **716** utilising the same operational functions invoked by processing nodes **706**,**711** in order to eventually match the motion of the real and composited planes with the composited clouds.

Upon completing the processing **813** at node **716**, the parent node **702** is thus able to pull all of the input data **801** to **813** and process it in order to generate a composite frame **814** comprising three elements, within which two planes appear superimposed over a sky and clouds. Sound data **815** will then be processed by node **703**, whereby output node **701** outputs final graphic data **814** and sound data **815** as an image frame **816**.

Figure 9

The main structure data **404** shown in *Figure 4* and further illustrated in *Figure 7* is shown in greater detail in *Figure 9* to better describe the hierarchy of data processing nodes it comprises as shown in *Figures 7* and *8*.

In the preferred embodiment of the present invention, each data processing node is referenced with a node ID **901** and is referenced as either a parent node **902**, a child node **903** or a combination thereof, as the case maybe. Each data processing node is further referenced by means of its type

data **405** at **904** and its function specific input data type **406** to **411** is similarly referenced at **905**.

The main structure **404** shown in *Figure 9* takes the form of a database for illustrative purposes only, as it will be apparent to those skilled in the art that any other referencing structure providing for the hierarchical structuring of data processing nodes is suitable.

In the preferred embodiment, the node ID **901** is generated during the first processing step **502** according to the processing order outlined in *Figures 7* and *8*, whereby the parent dependencies **902** and children dependencies **903** of each data processing nodes are generated and subsequently updated in real-time. Thus, in the example, the first node accessed at processing step **502** is the scene node **701** to which a node ID **906** is affixed. The next processing node accessed at said processing step **502** is data processing node **702** to which node ID **907** is affixed.

Data processing node **702** is a child node of data processing node **701**, whereby the parent dependency **908** of said parent node **701** is updated and the child dependency **909** of said data processing node **702** is similarly updated.

Thus, for each subsequent data processing node sequentially accessed during said processing step **502**, a new node ID is provided and the respective parent dependencies and child dependencies updated according to the principle outlined above, as are the corresponding node type **904** and node data type **905** populated, until such time as the entire main structure, or process tree, have been processed once. In the preferred embodiment of the present invention, one such processing cycle takes place within one thirtieth of a second.

Figure 10

Upon completing the processing step **502** and thus the registration of the parent and children data processing nodes as shown in *Figure 9* in order to establish their hierarchy, image data processing system **102** can provide a graphical representation of the image frame **603** in terms of a hierarchy of data processing nodes, which is shown in *Figure 10*.

In the example, user **101** provides first user input data **416** to the effect of activating user-operable switch **616**, the processing of which instructs image data processing system **102** to alternate the display mode in display area **602** from frame display mode to structure display **1001**, whereby said user operable switch **616** also alternates to user operable switch **1002** which, when activated, alternates said structure display **1001** back to frame display **602**, wherein data processing nodes **701** to **716** are processed to display rendered image components according to steps **502**, **503**.

In the example, the activation of user operable switch **616** generates structure display **1001**, whereby the entire main structure **404** is graphically represented, thus including data processing nodes **701** to **716** as a hierarchical structure of logically-linked data processing nodes, preferably but not necessary including a graphical representation of their corresponding respective output data **801** to **816**.

The method and apparatus as described above will so far be familiar to those skilled in the art, especially post-production image processing specialists. Said specialists will thus appreciate that the structure shown in *Figures 7, 8* and *10* is herein kept to an overly simplistic scale for the purpose of clarity in the present description as compared to traditional structures

comprising at the very least hundreds of such hierarchically structured data processing nodes. Consequently, said specialists will be well aware of the inherent difficulty in intuitively and rapidly identifying a particular data processing node, the parameters and/or data of which require editing, for instance to remove artefacts visible in image frame **603**, implement additional effects and/or components (such as add a third plane) or, more generally, simply to adjust the overall appearance of image frame **603** for aesthetic considerations.

According to the prior art, user **101** would activate the user operable switch **616** and thus be presented with the entire structure **404** defining image frame **603** as depicted within display area **1001**. User **101** would then have to firstly be familiar with the main structure **404** and its operating principle as described in *Figures 7* and *8*. Said user **101** would then have to identify the actual data processing nodes, the parameters and/or data of which require editing, from amongst all of said data processing nodes. Upon eventually identifying said relevant data processing node, said user would then select said node and amend the parameters and/or data thereof.

Thus, according to the known prior art, if user **101** needs to edit the positional data **409** processed by tracking node **711** in order to improve the motion of what is initially a three-dimensional model **804** within frame **603**, user **101** would have to identify and subsequently select said node **711** within structure **404** as displayed in GUI area **1001**.

Similarly, if user **101** needs to modify the artificial lighting applied to said model **804** at lighting node **709** using a light map **408**, user **101** would need to identify and subsequently select said lighting node **709** within the main structure **404** as shown in GUI area **1001**. It can thus be appreciated

that in an apparatus configured to the known prior art, or when using a method according to the known prior art, identifying and selecting a relevant data processing node from amongst hundreds or, more realistically, thousands of such logically-linked data processing nodes is unintuitive and time-consuming. Modem requirements for ever-more sophisticated effects applied to talents or environments or objects within a frame such as frame **603** compound the above problem, insofar as said effects require an ever increasing amount of varied data processing nodes.

Figure 11

The present invention remedies the above problem by providing a method of intuitively identifying and selecting any data processing node within main structure **404** with selectively culling parent dependencies **902** and children dependencies **903**. The processing steps according to which image data is edited at step **305** according to the present invention are further detailed in *Figure 11*.

In the example, user **101** is a post-production image editing specialist and wishes to improve the motion of the second background plane within frame **603**, initially generated from a three-dimensional model **406** together with its artificial lighting generated from a light map **408**. According to the invention, at step **1101** user **101** selects said second plane as an image component within image frame **603** within GUI **602**, whereby corresponding first user input data **416** is received and image processing application **402** firstly identifies all of the data processing nodes as sub-structures, e.g. layers within main structure **404**. First user input data **416** is then processed such that the image component of choice is selected at step **1102**.

According to the preferred embodiment of the invention, although user **101** in effect selects any which one of the plurality of data processing nodes, the processing of which generates said second plane within image frame **603**, it is irrelevant which specific data processing node within said sub-group is selected as application **402** eventually selects the first data processing node at step **1103** within the data processing node sub-structure identified at step **1102**.

In effect, at said step **1102**, application **402** identifies the last parent node of only the layer defining the image component selected at step **1101**, e.g. the first data processing node which defines a new layer or sub-structure after the keyer node **702**. In the example, said last data processing node is tracking node **711**.

A question is asked at the next step **1104** as to whether navigation input data has been received, i.e. whether application **402** has received further user input data **416** indicative of a decision that the data processing node **711** as selected at step **1103** is not the data processing node required for editing. In the preferred embodiment of the present invention, said navigation input data is provided by means of specific keystroke on keyboard **105** or mouse button activation on mouse **106**, but it will be apparent to those skilled in the art that any type of binary input device is appropriate to generate such navigation input data.

If the question of step **1104** is answered positively, a second question is asked at step **1105** as to whether the navigation input data received corresponds to linear navigating amongst parent and children nodes of a same layer, as first selected at step **1102**, or if it corresponds to transversal navigating amongst compatible children nodes of a different layer, whereby

said nodes are referred to as siblings.

If the question asked at step **1105** is answered positively, the next data processing node in the sub-structure identified at step **502** is selected at step **1107** which, in the example, is the scaling node **710**. Alternatively, the question asked at step **1105** is answered negatively, whereby a sibling is identified at step **1106** in a different sub-structure and control returned to question **1104**, such that navigation within said different sub-structure and node selection therein is now possible.

When the question asked at step **1104** is eventually answered negatively, a third question is asked at step **1108** as to whether application **402** has received selection input data. Said selection input data is again user input data **416** but differs from the navigation input data of question **1104** in that it is processed by application **402** for updating application data **406** to **411** to be processed by the data processing node selected at step **502**. Said selection input data is again read from keyboard **105** or mouse **106**, or a combination thereof but question **1108** is only answered positively if said input data differs from the pre-set navigation input data of questions **1104**, e.g. input data read at question **1108** differs from the keystroke or mouse click input to effect navigation at question **1104**.

Thus, when question **1108** is answered positively, the editing of parameters and/or data is initiated at step **1109** for the data processing node selected at step **1107**, and the edited image data may eventually be stored according to step **306**. Alternatively, question **1108** is answered negatively whereby the edited image data may immediately be stored according to step **306**.

Figure 11b

The operational steps according to which sub-structures of data processing nodes are identified at step **1101** are further described in *Figure 11b*. At step **1111**, recurring child node references are identified within the main structure as repeated occurrences of the dependency of a data processing node on a parent data processing node. Thus, in the example, the first level of substructural division is based upon the dependency of image keying node **702** shown at **907** with a child node reference **909** and sound mixing node **703** with a node ID **901** equal to 0016 and an identical child node reference equal to 0001 respectively children of seen output node **701** shown at **906**.

Further in the example, the editor is only concerned with image data and thus the next level of substructural division is attained by identifying the recurring graphic child node reference **903** equal to 0002.

Upon completing the above identification step, a boundary is set for each substructure, or layer, at step **1112** based upon said identified recurring graphic child node reference, whereby a precise identification of each substructure presence within main structure **404** is made possible and accomplished at step **1113**. Referring back to *Figure 9*, four substructures each defining a separate image component are identified, the respective first parent node of which are tracking node **706** (node ID 0003), tracking node **711** (node ID 0006), colour correction node **713** (node ID 0011) and tracking node **716** (node ID 0013).

Figure 11c

The operational steps according to which an image component is selected by means of its corresponding sub-structure at step **1102** are further described in *Figure 11c*. At step **1121**, the X, Y input data **416** read at step **1101** is processed in order to identify which substructure defines the selected image component. It was previously explained at that the precise identification of a specific data processing node within the substructure is irrelevant as it is the ultimate parent node within the substructure which is eventually selected by means of processing the nodes dependencies within said substructure. Thus, as the X, Y input data **416** is processed to identify a specific node, its ultimate parent node is eventually identified as shown at **1121**, wherein in the example the lighting node **709** (node ID 0008) is derived from said input data processing, its dependencies processed and its ultimate parent node tracking node **711** (node ID 0006) selected thus identifying the selected image component as belonging to layer two.

Upon completing the above identification step, all of the data processing nodes of the identified substructure are selected to the exclusion of any other data processing node belonging to other substructures.

Figure 11d

The operational steps according to which compatible data processing nodes are identified and selected in a different sub-structure at step **1106** are further described in *Figure 11d*. At step **1131**, the node type **904** of the data processing nodes selected at step **1107** is read in main structure **404** and the next substructure is selected at step **1132**, preferably by identifying the ultimate parent node of the current substructure and

identifying the ultimate parent node of the next substructure.

At step **1133**, the equivalent data processing node is selected in the new layer selected at step **1132**, preferably by means of processing the same number of dependencies, whereby the respective node type **904** of the initial data processing node read at step **1131** and the sibling data processing node read at step **1133** are compared for type match at step **1134**. A question is subsequently asked at step **1135** in order to determine if said respective mode types **904** match. If the question of step **1135** is answered negatively, the next dependant node will be selected at step **1133** for node type matching at step **1134**, for instance if various other functionality's are invoked by means of data processing nodes with a different node type between the substructure defining node and the sibling node in said next layer. The question of step **1135** is eventually answered positively, whereby control returns to the question of step **1104**, e.g. whether navigation input data has been received or not.

Figure 12

The graphical user interface **601** of application **402** is shown in *Figure 12*, wherein user **101** has selected the second plane as an image component of image frame **603** according to step **1101**.

In the preferred embodiment of the present invention, the data processing nodes **707**, **708**, **709** and **711** are identified at step **1101** as the sub-structure of nodes generating the second plane as an image component of image frame **603** and the tracking node **711** is selected as the ultimate processing node applied to said image component according to step **1103**.

Consequently, until such time as application **402** receives either navigation input data, the effect of which would be to select the next node **710** within said sub-structure, or selection input data, the effect of which would be to initiate the editing of the current parameters and/or data of tracking node **711**, a graphical representation of said parameters and data specific to the functionality of the current selected node **711** is provided within image data **603**. Said representation includes a vector **1201**, the direction of which depicts the direction of travel of the plane within a reference three-dimensional space **1202** and the length of which depicts the velocity of said plane.

A three-dimensional cubic volume **1203** encompasses the plane within said three-dimensional reference space **1202** to depict more intuitively the offset angle between the relative floor of space **1202** as defined by abscisses X and Z and the wing span of the three-dimensional model corresponding to said plane.

According to the invention, navigation user input data may now be provided as an answer to question **1104**, preferably within the same sub-structure as an answer to question **1105**, whereby parameters and data combinations **1201** to **1203** specifically relating to the functionality of tracking node **711** would be replaced by graphical representation of parameters and data combinations specific to the next data processing structure **710**, ie specific to the editing of parameters and/or data relating to the scaling of the plane within the reference three-dimensional space **1202** and so on and so forth until the process selects the last child node **707** within said sub-structure, whereby editing of parameters and/or data could for instance include the generation of a completely different type of plane as an

alternative three-dimensional model **406**.

Figure 13

According to another preferred embodiment of the present invention, the user **101** of image processing system **102** configured according to the invention may prefer to edit the parameters and/or data graphically represented at **1201** to **1203** in more detail than can be perceived as graphically depicted in GUI **601** as shown in *Figure 12* and thus application **402** may receive user input data **416** to the effect that the user operable switch **616** is activated such that the sub-structure of data processing nodes identified according to step **1102** is displayed within structure display area **1001**. A graphical representation of said structure display area **1001** configured according to the present invention is shown in *Figure 13*.

In the example, user **101** selects the second plane as described in *Figure 12* but subsequently activates user operable switch **616** before inputting navigation input data according to question **1104** or selection input data according to question **1108**, whereby frame display area **602** alternates to structure display area **1001** and said user operable switch **616** alternates to user operable switch **1002**. The main structure **404** is culled according to step **1122** and the ultimate data processing node is selected in the sub-structure of data processing nodes defining said selected second plane, whereby only said sub-structure is graphically displayed at **1301**, which only comprises data processing nodes **711** to **707** in accordance with the example previously described in *Figure 12*.

In the alternative preferred embodiment, the selected first data processing node **711** is highlighted such that image editor **101** is immediately

made aware of which type of parameters and/or data he may initiate the editing thereof according to step **1109**. Said data processing node highlighting is provided here only as an example and it will be readily apparent to those skilled in the art that any other alternative providing a similar functionality is here envisaged.

In the alternative embodiment still, application **402** generates a parameters and data display area **1302** corresponding to said selected data processing node which, in the example is tracking node **711**. Depending upon the processing functionality provided by said selective node, said parameters and data display area **1302** may vary to a large extent. Preferably, area **1302** clearly identifies tracking node **711** by means of its corresponding node type data **405** referenced at **904**, as well as its various parameters **1303** and parameter data **1304**.

In the alternative preferred embodiment of the present invention, user **101** may input selection input data according to question **1108** within parameter data **1304**, within which data corresponding to graphical representations **1201** to **1203** are shown for the purpose of clarity. Application **402** may at any time receive user input data **416** to the effect of either editing parameter **1303** and/or parameter data **1304**, or to the effect of navigating further within sub-structure **1301** in accordance with processing steps described in *Figure 11*, or to the effect that user operable switch **1002** has been activated and this alternates structure display area **1001** back to image frame display area **602**, the generation of the image components of which would thus incorporate any editing implemented within parameters **1303** and/or parameter data **1304**.

Figure 14

The graphical user interface **601** of application **402** is shown in *Figure 14*, wherein user **101** has selected the second plane as an image component of image frame **603** according to step **1101** and subsequently provided application **402** with navigation input data according to question **1104** in order to edit the lighting data of lighting node **709**.

It was previously explained that the data processing nodes **707**, **708**, **709**, **710** and **711** are identified at step **1101** as the sub-structure of nodes generating the second plane as an image component of image frame **603** and the tracking node **711** is selected as the ultimate processing node applied to said image component according to step **1103**.

Consequently, when application **402** receives navigation input data, the effect of which is to select the next node **709** within said sub-structure, a graphical representation of said parameters and data specific to the functionality of the current selected node **709** is provided within image data **603**. Said representation includes a spotlight **1401** indicative of the type of lightmap **408** said second plane is processed with. Said representation also includes a focus area **1402** and corresponding light cone **1403** respectively depicting the focus of spotlight **1401** and the lighting energy provided thereto, within the reference three-dimensional space **1202**. Finally, said representation includes a user-operable colour selector **1404** with which to intuitively select appropriate RGB colour data for said lightmap. User 101 has therefore rapidly identified the relevant data processing node and may now input selection input data according to steps **1108** and **1109**, thus intuitively editing parameters and/or data to be processed by the selected lighting node

709 to improve the characteristics of the selected image component.

Figure 15

According to another preferred embodiment of the present invention, the user **101** of image processing system **102** configured according to the invention may prefer to edit the parameters and/or data graphically represented at **1401** to **1404** in more detail than can be perceived as graphically depicted in *Figure 14* and thus application **402** may receive user input data **416** to the effect that the user operable switch **616** is activated such that the sub-structure of data processing nodes identified according to step **1101** is displayed within structure display area **1001**. A graphical representation of the structure display area **1001** depicting the image data editing shown in *Figure 14* is shown in *Figure 15*.

In the example, user **101** again selects the second plane as described in *Figure 12* but subsequently activates user operable switch **616** before inputting navigation input data according to question **1104**, whereby frame display area **602** again alternates to structure display area **1001** and said user operable switch **616** alternates to user operable switch **1002**. The main structure **404** is culled according to step **1122** and the ultimate data processing node is selected in the sub-structure of data processing nodes defining said selected second plane, whereby only said sub-structure is graphically displayed at **1501**, which only comprises data processing nodes **707** to **711** in accordance with the example previously described.

In the alternative embodiment, the selected last data processing node **711** is highlighted such that image editor **101** is immediately made aware of which type of parameters and/or data he may initiate the editing thereof

according to step **1109**. User **101** however wishes to edit the parameter data of lighting node **709** and thus provides application **402** with navigation input data at **1104** until such time as said data processing node **709** is highlighted.

In the alternative preferred embodiment still, application **402** again generates a parameters and data display area **1302** corresponding to said selected data processing node **709**. As the processing functionality provided by said selected node differs from the processing functionality of node **711**, said parameters and data display area **1302** vary to a large extent relative to that shown in *Figure 13*. Preferably, area **1302** clearly identifies tracking node **709** by means of its corresponding node type data **405** referenced at **904**, as well as its various parameters **1502** and parameter data **1503**.

In the alternative preferred embodiment of the present invention, user **101** may input selection input data according to question **1108** within parameter data **1503**, within which data corresponding to graphical representations **1401** to **1404** are shown for the purpose of clarity. Application **402** may at any time receive user input data **416** to the effect of either editing parameters **1502** and/or parameter data **1503**, or to the effect of further navigating within sub-structure **1501** in accordance with processing steps described in *Figures 11* to *14*, or to the effect that user operable switch **1002** has been activated and this alternates structure display area **1001** back to image frame display area **602**, the generation of the image components of which would thus incorporate any parameters and/or data editing implemented within parameters **1502** and/or parameter data **1503**.

Figure 16

The graphical user interface **601** of application **402** is shown in *Figure 16*, wherein user **101** has selected the tracking node **706** eventually applied to frame data **801** as a sibling of tracking node **711** eventually applied to 3D model data **804**, according to step **1106**.

The provision for transversal navigation allows the editor **101** to intuitively and rapidly edit the data and/or parameters of a plurality of data processing nodes respectively belonging to separate sub-structures, the functionality of which is comparable. In the example, editor **101** has edited parameter data **1304** processed by tracking node **711** in frame view, but now wishes to edit the similar parameter data processed by tracking node **706** to fine-tune the relative positioning between the film-based plane and the 3D-modelled plane in the output image frame. Upon accomplishing the procedural steps described in *Figure 12*, the currently selected data processing node is therefore tracking node **711**.

Upon inputting corresponding navigation input data **416** interpreted at question **1105** as selecting a different sub-structure, data processing node **716** is identified as the next sibling node featuring comparable data type **904** in main structure **404**, which motion-tracks the 3D-modelled clouds. But editor **101** wishes to edit data pertaining to the generation of the first plane, therefore further navigation input data **416** is input to be interpreted at question **1105** as selecting a different sub-structure, whereby data processing node **706** is identified as the next sibling node featuring comparable data type **904** in main structure **404**, which orients the plane frame **801** within the three-dimensional reference space **1202**.

The representation of the functionality, parameters and data of tracking node **706** includes a two-dimensional plane **1601** encompassing the

frame within said three-dimensional reference space **1202** to depict more intuitively the offset angle between the relative floor of space **1202** as defined by abscisses X and Z and the frame **801**. Said representation also includes a vector **1602**, the direction of which depicts the direction of travel of the frame within said reference three-dimensional space **1202** and the length of which depicts the velocity of said frame.

According to the invention, navigation user input may now be provided as an answer to question **1104**, whereby parameters and data combinations **1201** to **1203** specifically relating to the functionality of tracking node **706** would be replaced by graphical representation of parameters and data combinations specific to the next data processing nodes **705** or **704**, ie specific to the editing of parameters and/or data relating to the colour of the frame within the reference three-dimensional space **1202** or specific to the generation of a completely different frame.

Figure 17

According to another preferred embodiment of the present invention, the user **101** of image processing system **102** configured according to the invention may prefer to edit the parameters and/or data graphically represented at **1601** and **1602** in more detail than can be perceived as graphically depicted in *Figure 16* and thus application **402** may receive user input data **416** to the effect that the user operable switch **616** is activated such that the sub-structure of data processing nodes identified according to step **1101** is displayed within structure display area **1001**. A graphical representation of said structure display area **1001** configured according to the present invention is shown in *Figure 17*.

In the example, user **101** selects the second plane as described in *Figure 12* but subsequently activates user operable switch **616**, whereby sub-structure **1301** is initially displayed as only comprising data processing nodes **707** to **711** in accordance with the example previously described in *Figure 13*. However, user **101** wishes to select sibling node **706** as described in *Figure 16* and thus imparts navigation input data interpreted at question **1105** as navigating in a different sub-structure.

The main structure **404** is therefore eventually culled according to step **1122** when data processing node **706** is selected as heading the next sub-structure **1701** at step **1132**, and further highlighted when selected as sibling because its data type is matched against the data type of node **711** according to step **1135**. Image editor **101** is again immediately made aware of which type of parameters and/or data he may initiate the editing thereof according to step **1109**. Moreover, image editor **101** is also immediately aware that edits will be performed for a different data processing node applied to a different image component, as the displayed topography of the sub-structure has changed.

Application **402** again generates a parameters and data display area **1702** corresponding to said selected data processing node **706**. User **101** may input selection input data according to question **1108** within parameter **1703**, within which data **1704** corresponding to graphical representations **1601** and **1602** are shown for the purpose of clarity. Application **402** may at any time receive user input data **416** to the effect of either editing parameter parameters **1703** and/or parameter data **1704**, or to the effect of navigating further within sub-structure **1701** in accordance with processing steps described in *Figure 11*, or to the effect that user operable switch **1002** has

been activated and this alternates structure display area **1001** back to image frame display area **602**, the generation of the image components of which would thus incorporate any editing implemented within parameters **1703** and/or parameter data **1704**.

Claims:

1. Apparatus for processing image data comprising storage means, processing means and display means, wherein said image data includes a plurality of components defined by a hierarchy of data processing nodes, said storage means are configured to store said image data and said processing means are configured to perform the steps of

generating first image data by means of processing said hierarchy of data processing nodes;

outputting said first image data to said display means;

restructuring said hierarchy in response to user input data;

generating second image data by means of processing said restructured hierarchical processing nodes; and

outputting said second image data to said display means.

2. Apparatus according to claim 1, wherein said hierarchy organises said data processing nodes vertically as parent nodes, children nodes and sibling nodes.

3. Apparatus according to claim 2, wherein said sibling nodes define horizontal layers within said hierarchy.

4. Apparatus according to any of claims 1 to 3, wherein said user input data includes selection data indicating the selection of one or a plurality of said components.

5. Apparatus according to any of claims 1 to 3, wherein said user input data includes navigation data indicating navigation within said hierarchy or a portion thereof.

6. Apparatus according to claim 5, wherein said navigation is either horizontal or vertical.

7. Apparatus according to any of claims 1 to 6, wherein said restructuring step includes culling said hierarchy down to one or a plurality of portions thereof.

8. Apparatus according to claim 7, wherein said second image data is a representation of said portion or portions.

9. Apparatus according to any of claim 1 to 7, wherein said second image data is an image frame.

10. A method of processing image data, wherein said image data includes a plurality of components defined by a plurality of hierarchical data processing nodes, said method comprising display means and further comprising the steps of

generating first image data by means of processing said hierarchical data processing nodes;

outputting said first image data to said display means;

restructuring said hierarchy in response to user input data;

generating second image data by means of processing said restructured hierarchical processing nodes; and

outputting said second image data to said display means.

12. A method according to claim 11, wherein said hierarchy organises said data processing nodes vertically as parent nodes, children nodes and sibling nodes.

13. A method according to claim 12, wherein said sibling nodes define horizontal layers within said hierarchy.

14. A method according to any of claims 11 to 13, wherein said user input data includes selection data indicating the selection of one or a plurality of said components.

15. A method according to any of claims 11 to 13, wherein said user input data includes navigation data indicating navigation within said hierarchy or a portion thereof.

16. A method according to claim 15, wherein said navigation is either horizontal or vertical.

17. A method according to any of claims 11 to 16, wherein said restructuring step includes culling said hierarchy down to one or a plurality of portions thereof.

18. A method according to claim **17**, wherein said second image data is a representation of said portion or portions.

19. A method according to any of claim **11** to **17**, wherein said second image data is an image frame.

20. A computer-readable medium having computer-readable instructions executable by a computer, such that said computer performs the steps of:

generating first image data by means of processing hierarchical data processing nodes defining said image data;

outputting said first image data to display means;

restructuring said hierarchy in response to user input data;

generating second image data by means of processing said restructured hierarchical processing nodes; and

outputting said second image data to said display means.

21. A method according to claim **20**, wherein said user input data includes selection data indicating the selection of one or a plurality of said components.

22. A method according to any of claim **20**, wherein said user input data includes navigation data indicating navigation within said hierarchy or a portion thereof.

23. A method according to any of claims **22**, wherein said restructuring step includes culling said hierarchy down to one or a plurality of said portions.

24. A method according to claim **23**, wherein said second image data is a representation of said portion or portions.

25. A method according to claim **23**, wherein said second image data is an image frame.

26. A computer system programmed to generate image data, storage means, processing means and display means, wherein said image data includes a plurality of components defined by a plurality of hierarchical data processing nodes, said storage means are configured to store said image data and said processing means are configured to perform the steps of

generating first image data by means of processing said hierarchical data processing nodes;

outputting said first image data to said display means;

restructuring said hierarchy in response to user input data;

generating second image data by means of processing said restructured hierarchical processing nodes; and

outputting said second image data to said display means.

Abstract of the Disclosure

Editing Image Data

An apparatus and method for processing image data **403** are provided, which comprise storage means **206**, processing means **201**, 202 and display means **103**, wherein said image data **403** includes a plurality of components **814** defined by a plurality of hierarchical data processing nodes **404**, **405**, said storage means are configured to store said image data **403** and said processing means are configured to perform the steps of generating first image data **1201**, **1202**, **1203** by means of processing said hierarchical data processing nodes **405**; outputting said first image data to said display means **103**; restructuring **(1122)** said hierarchy **404** in response to user input data; and generating second image data **1401**, **1402**, **1403**, **1404** by means of processing said restructured hierarchical editing nodes **405**; and outputting said second image data to said display means.

(Figure 11)

2034-P578-GB

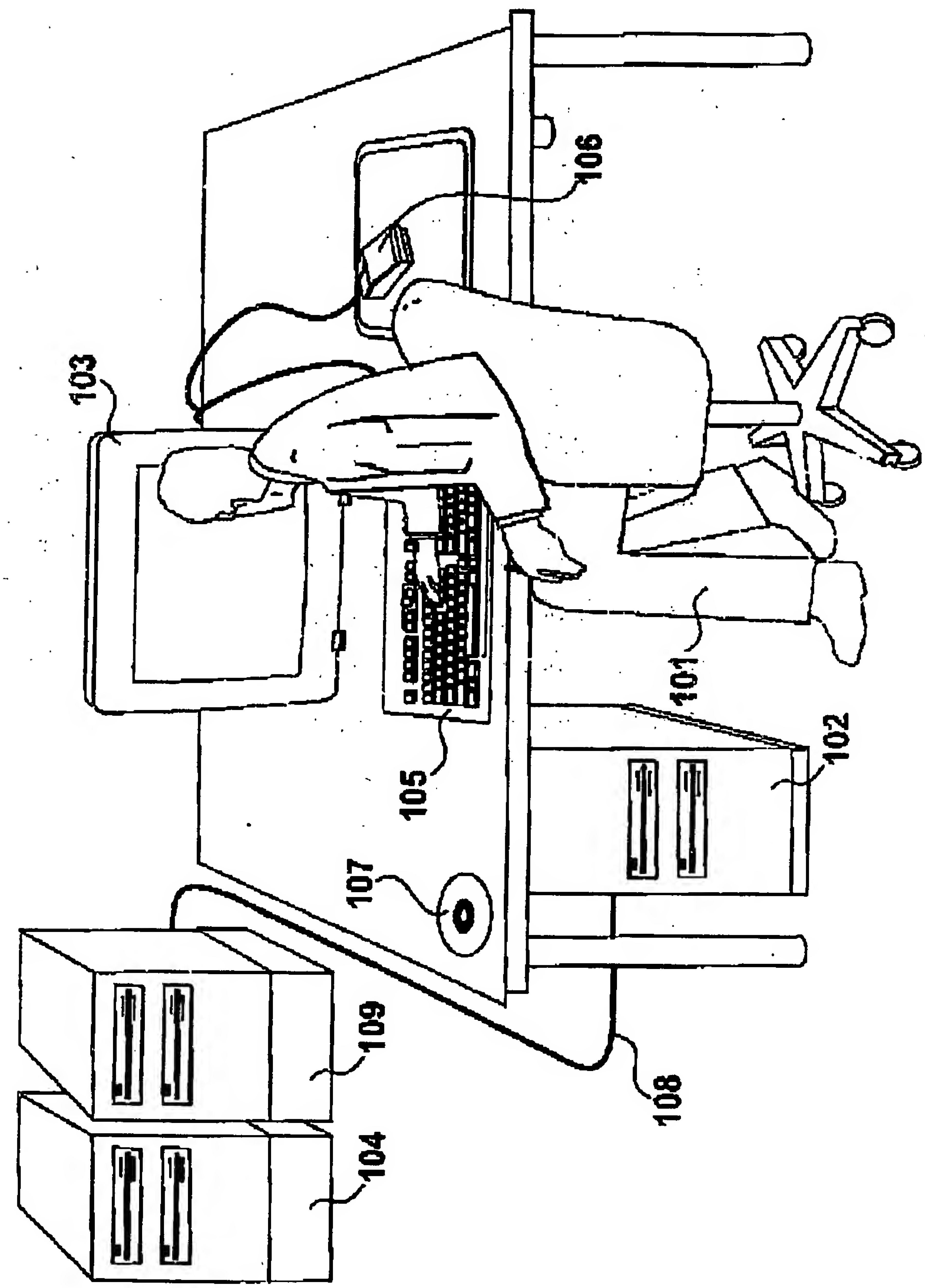


Figure 1

2034-P578-GB

(

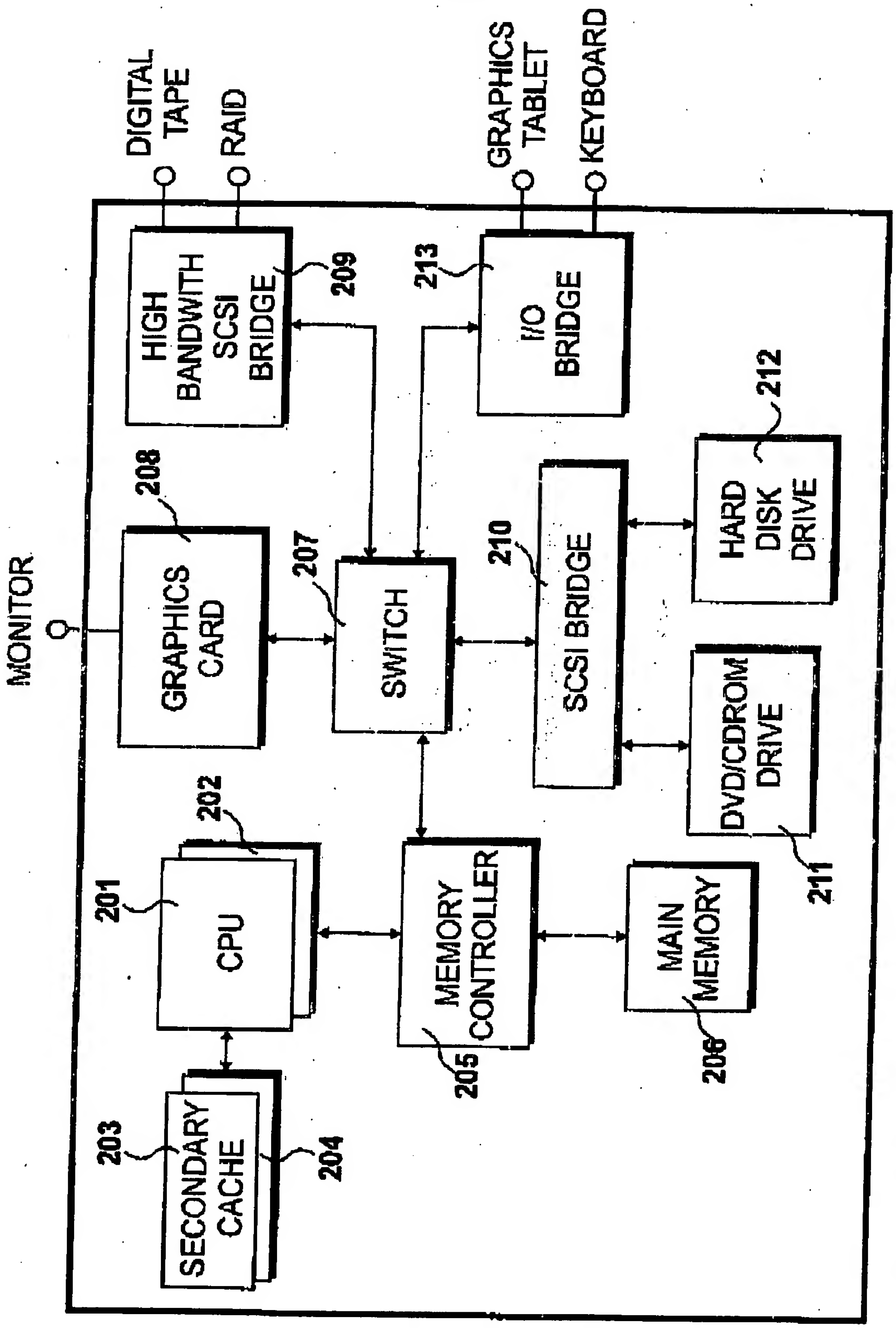


Figure 2

3/20

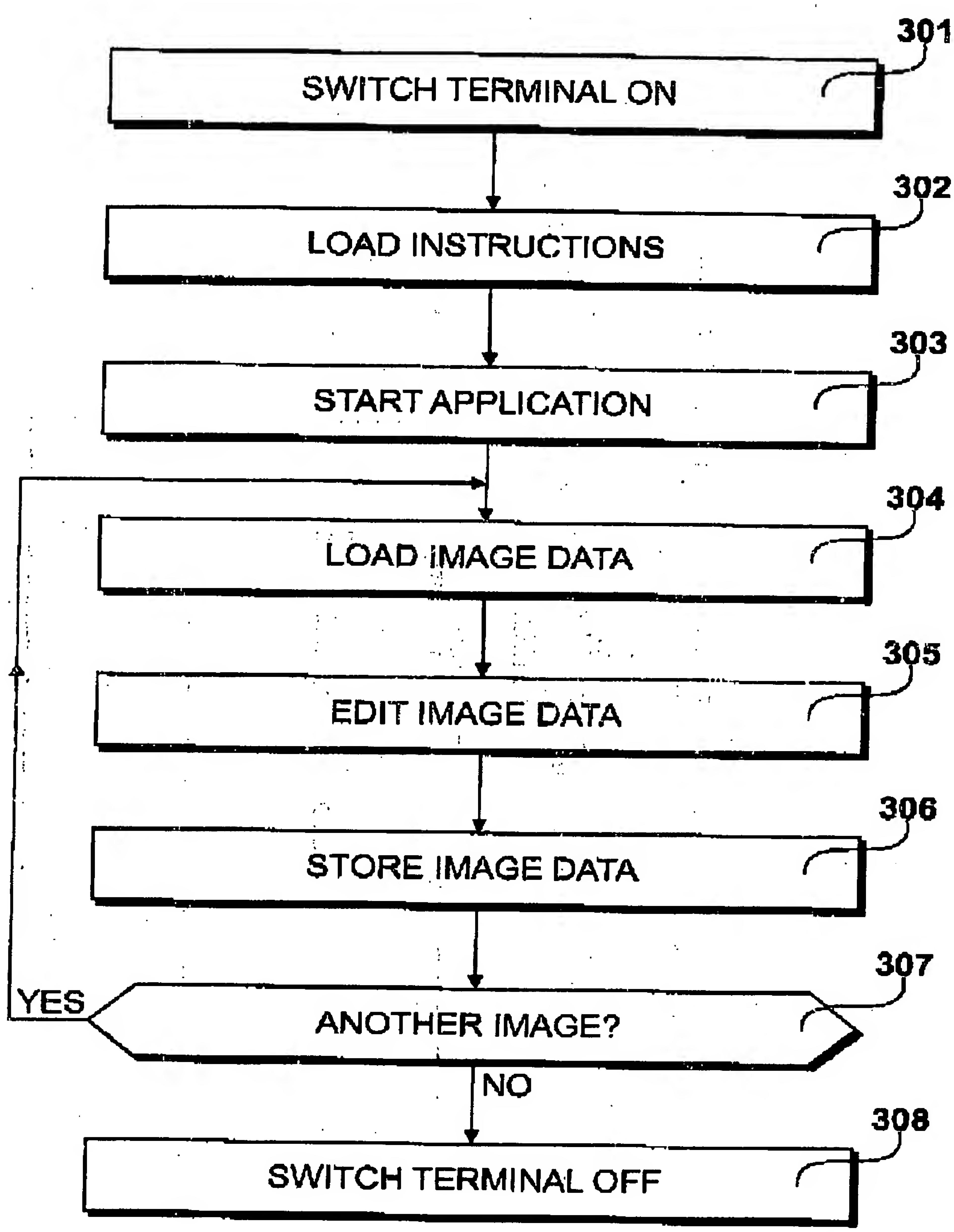

301
SWITCH TERMINAL ON
302
LOAD INSTRUCTIONS
303
START APPLICATION
304
LOAD IMAGE DATA
305
EDIT IMAGE DATA
306
STORE IMAGE DATA
307
YES
ANOTHER IMAGE?
NO
308
SWITCH TERMINAL OFF

Figure 3

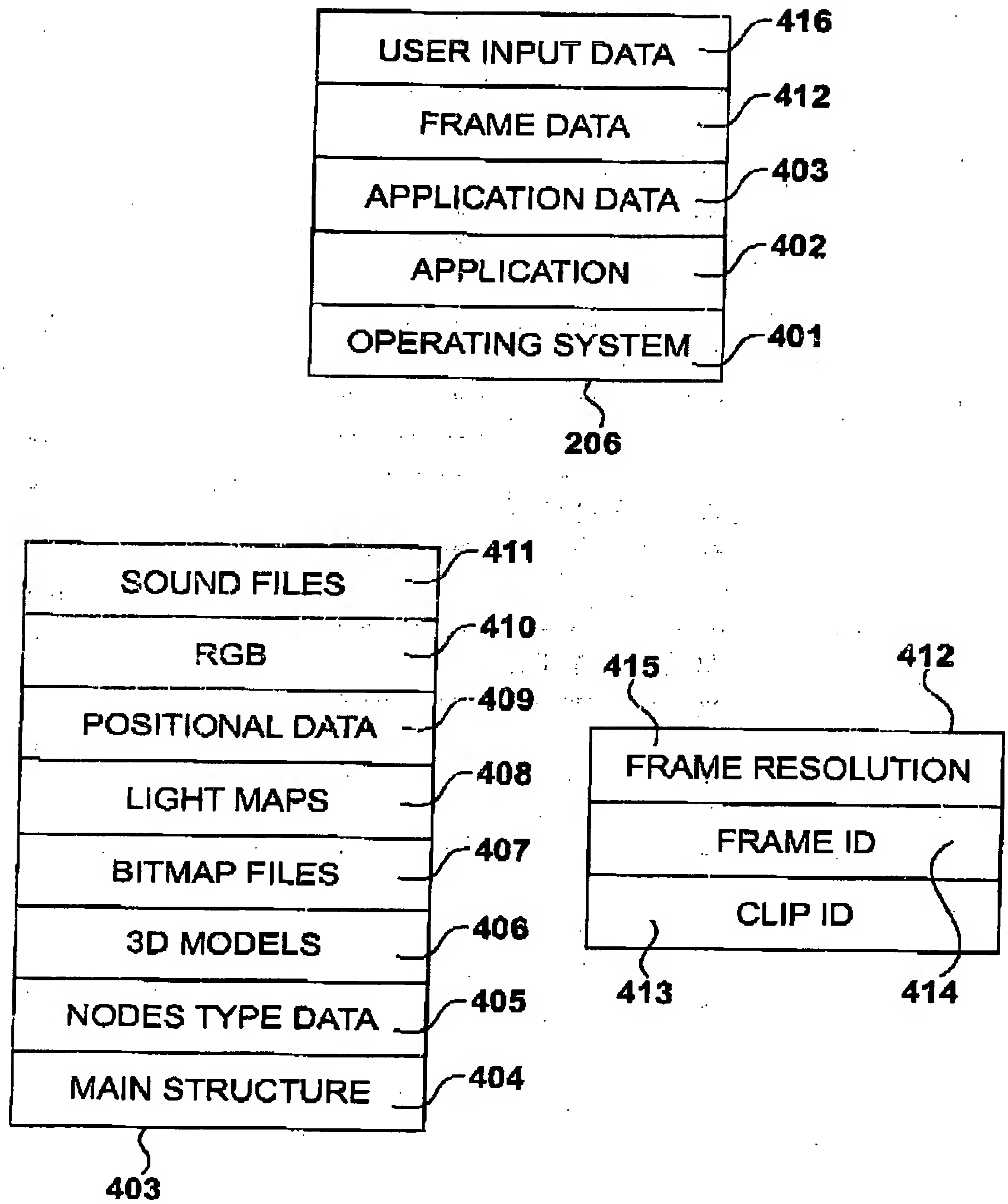

USER INPUT DATA
416
FRAME DATA
412
APPLICATION DATA
403
APPLICATION
402
OPERATING SYSTEM
401
206
SOUND FILES
411
RGB
410
POSITIONAL DATA
409
LIGHT MAPS
408
BITMAP FILES
407
3D MODELS
406
NODES TYPE DATA
405
MAIN STRUCTURE
404
403
415
412
FRAME RESOLUTION
FRAME ID
CLIP ID
413
414

Figure 4

2034-P578-GB

5/20

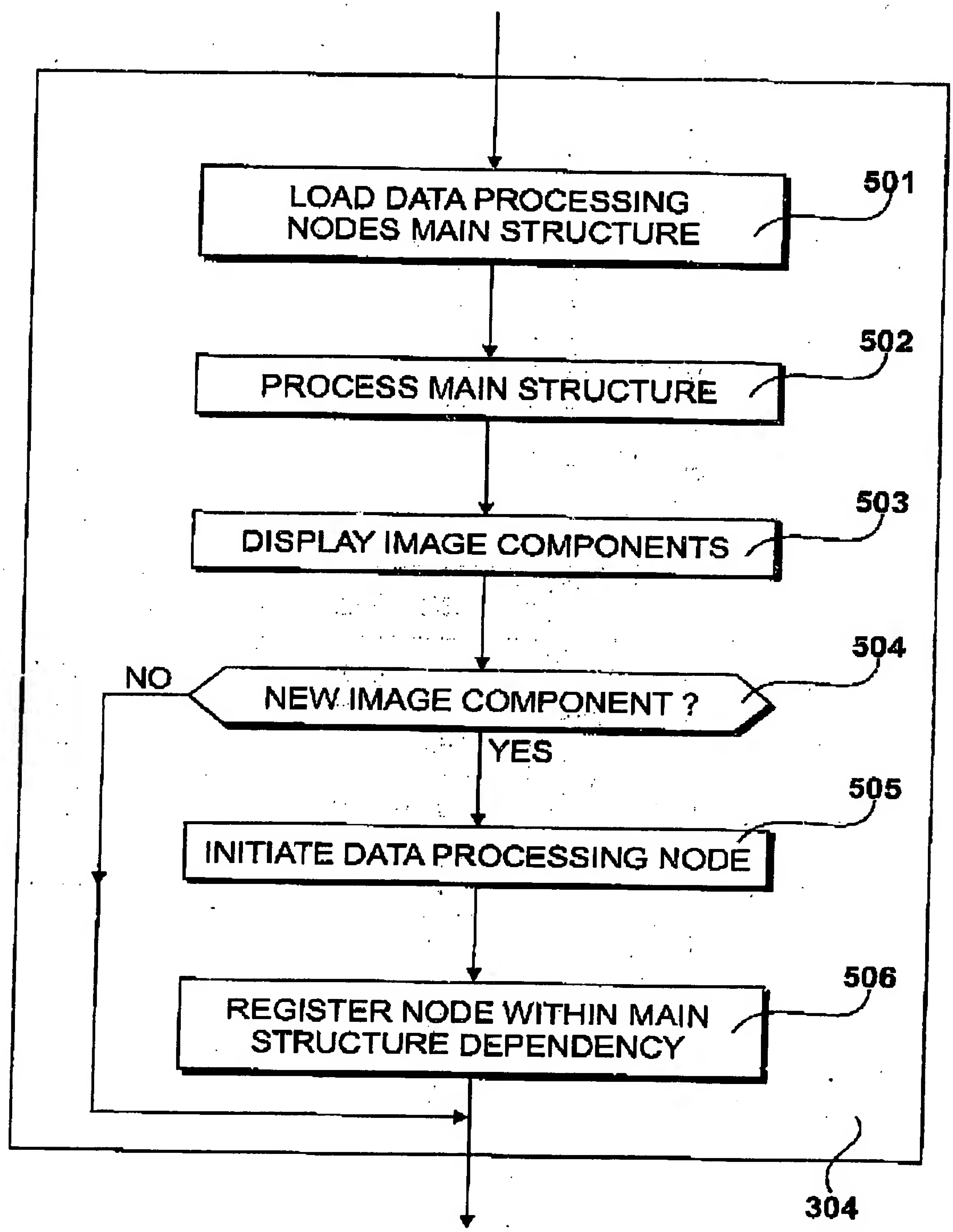


Figure 5

2034-P578-GB

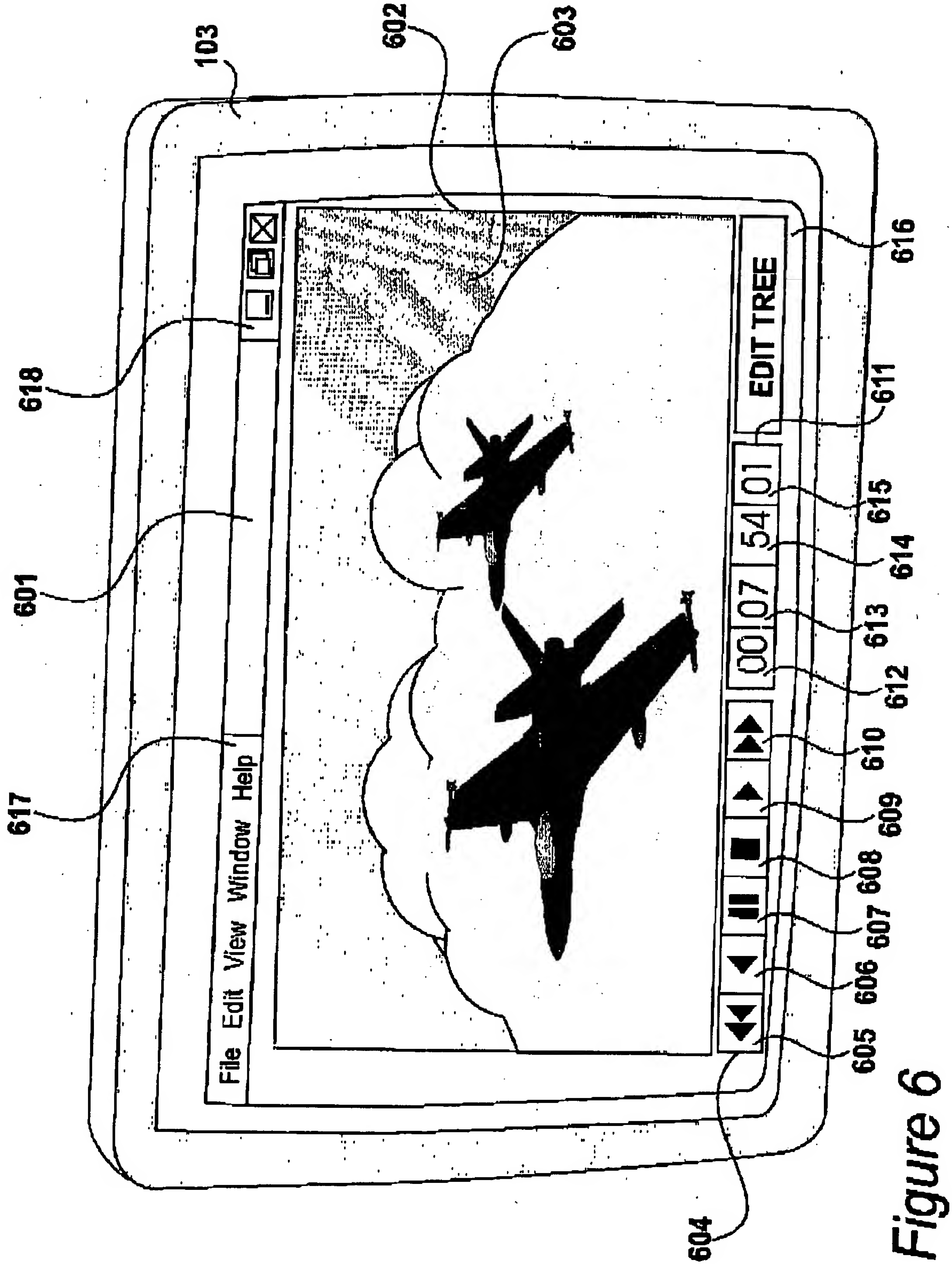
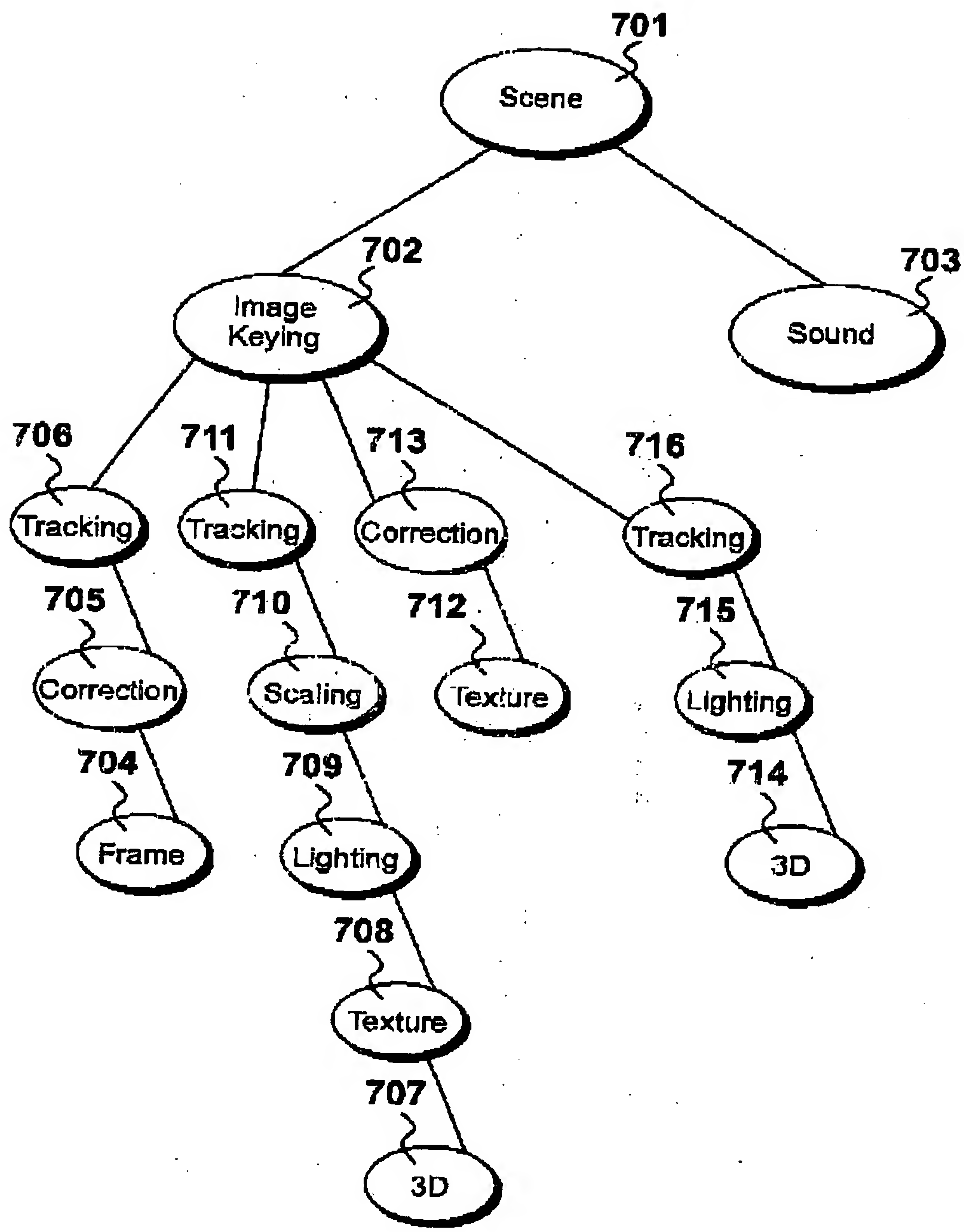


Figure 6



701
Scene
702
Image Keying
703
Sound
706
Tracking
711
Tracking
713
Correction
716
Tracking
705
Correction
710
Scaling
712
Texture
715
Lighting
704
Frame
709
Lighting
714
3D
708
Texture
707
3D

Figure 7

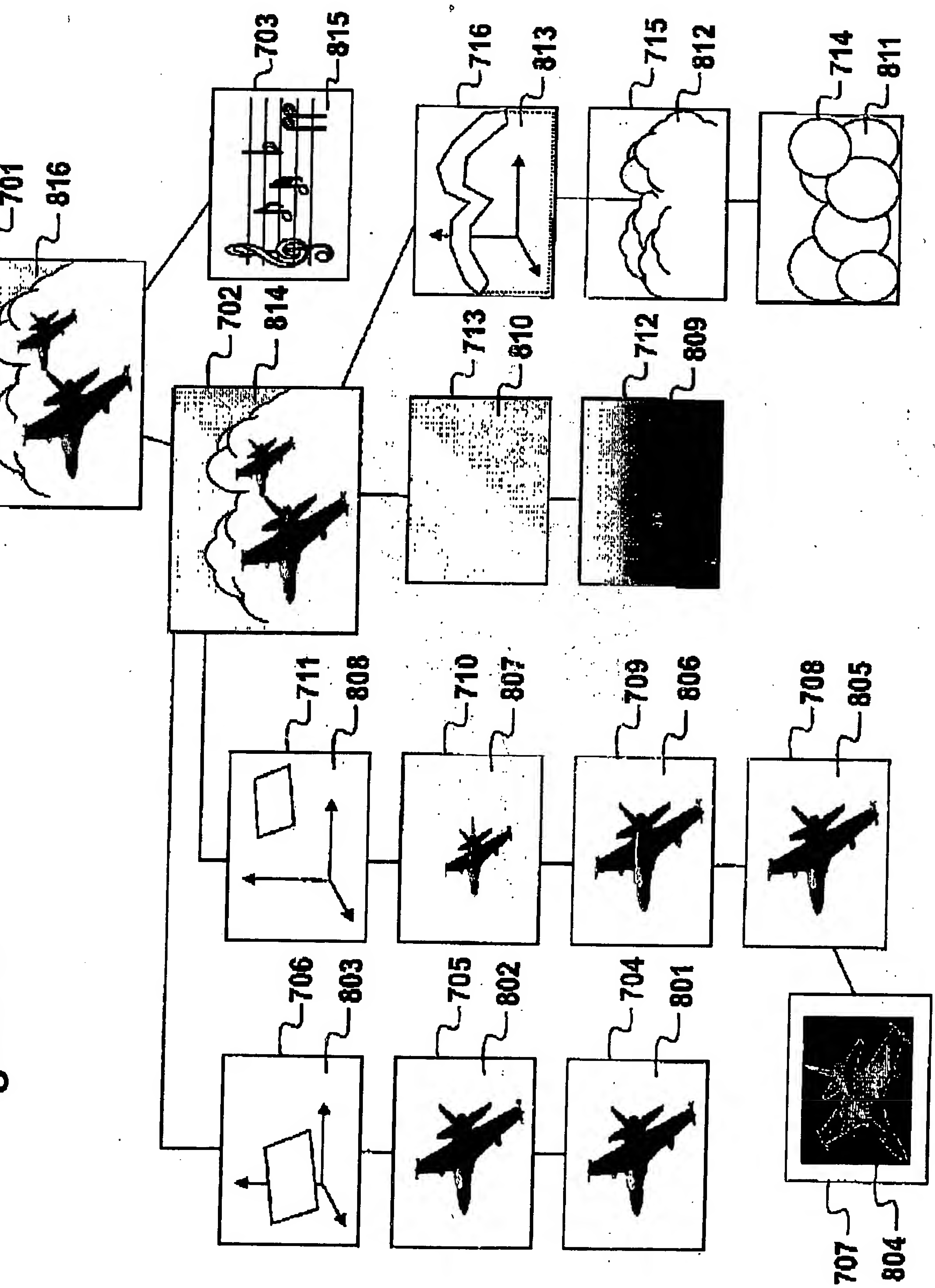

701
816
703
815
702
814
716
813
715
812
714
811
713
810
712
809
711
808
710
807
709
806
708
805
706
803
705
802
704
801
707
804

Figure 8

901 908 902 909 903 904 905

NODE ID	PARENT	CHILD	NODE TYPE	NODE DATA
0001	0002:0016	—	SCENE OUTPUT	ALL
0002	0003:0015	0001	KEYER	RGB
0003	0004:0005	0002	TRACKER	VECTOR
0004	0005	0003	SUPPRESS	RGB
0005	—	0004	FRAME	RGB
0006	0007:0010	0002	TRACKER	VECTOR
0007	0008:0010	0006	SCALE	VECTOR
0008	0009:0010	0007	LIGHTING	LIGHT MAP
0009	0010	0008	TEXTURE	BITMAP
0010	—	0009	MODELER	3D MODEL
0011	0012	0002	SUPPRESS	RGB
0012	—	0011	TEXTURE	BITMAP
0013	0014:0015	0002	TRACKER	VECTOR
0014	0015	0013	LIGHTING	LIGHT MAP
0015	—	0014	MODELER	3D MODEL
0016	—	0001	MIXER	SOUND FILE

906 907 404

Figure 9

2034-P578-GB

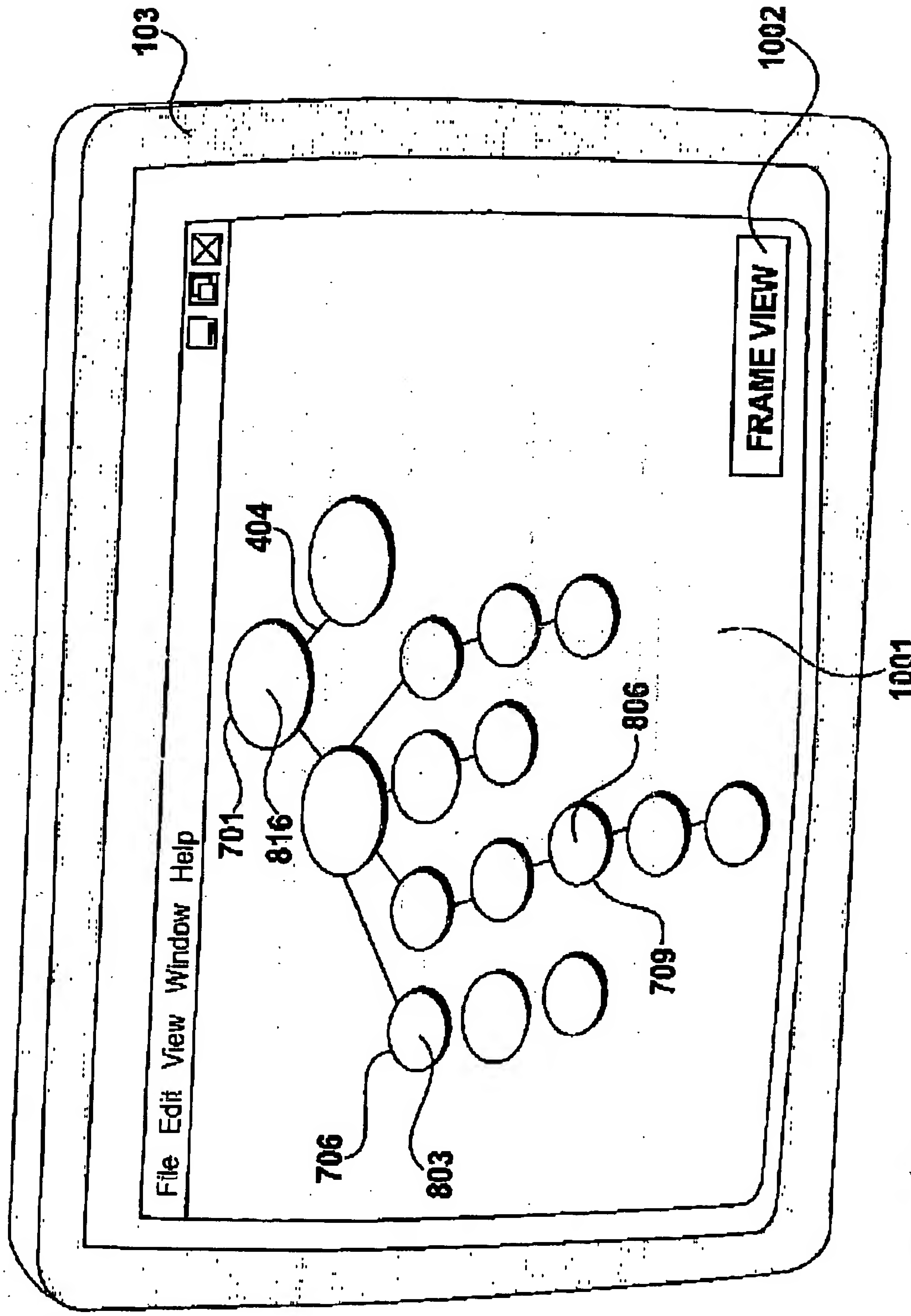


Figure 10

2034-P578-GB

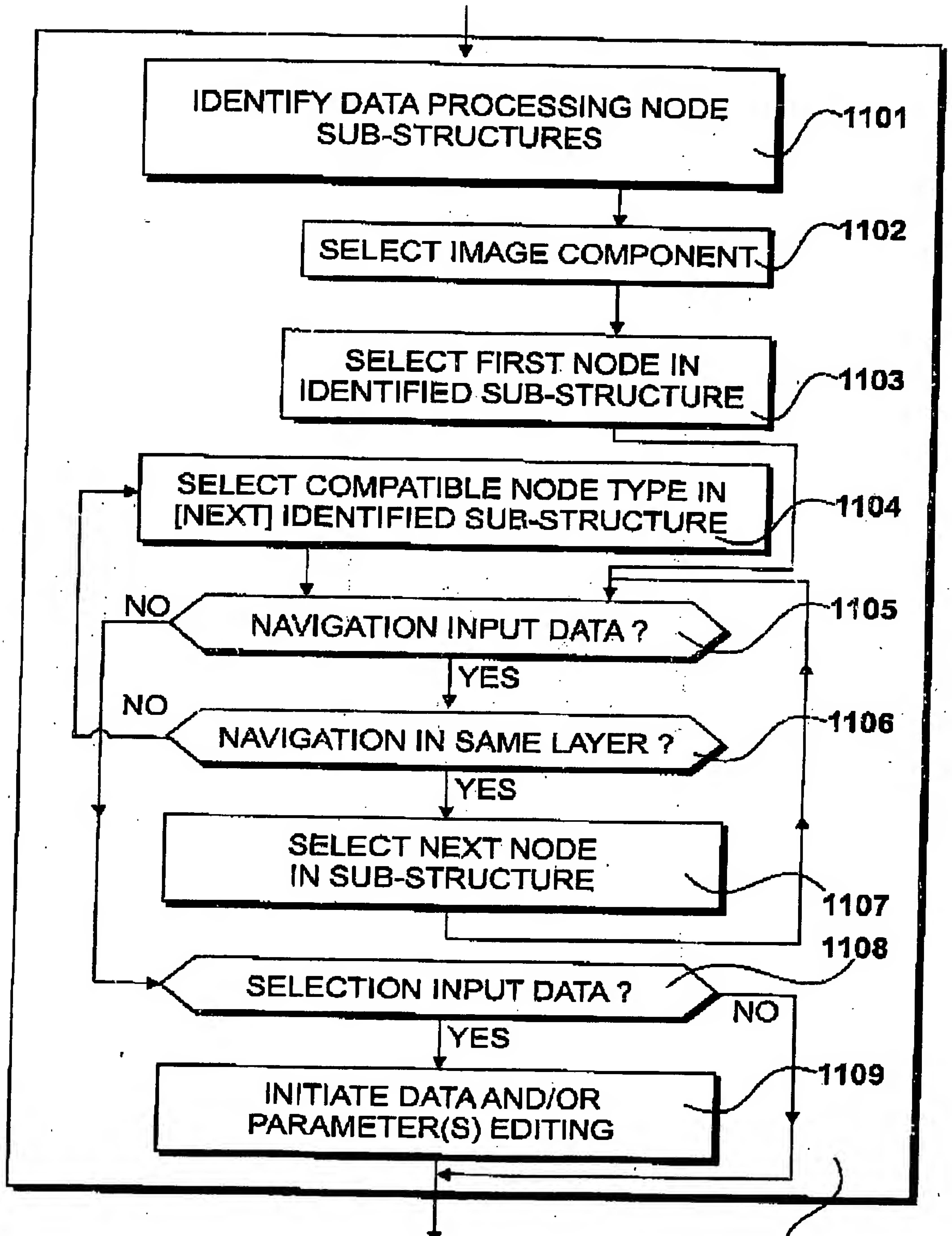


Figure 11

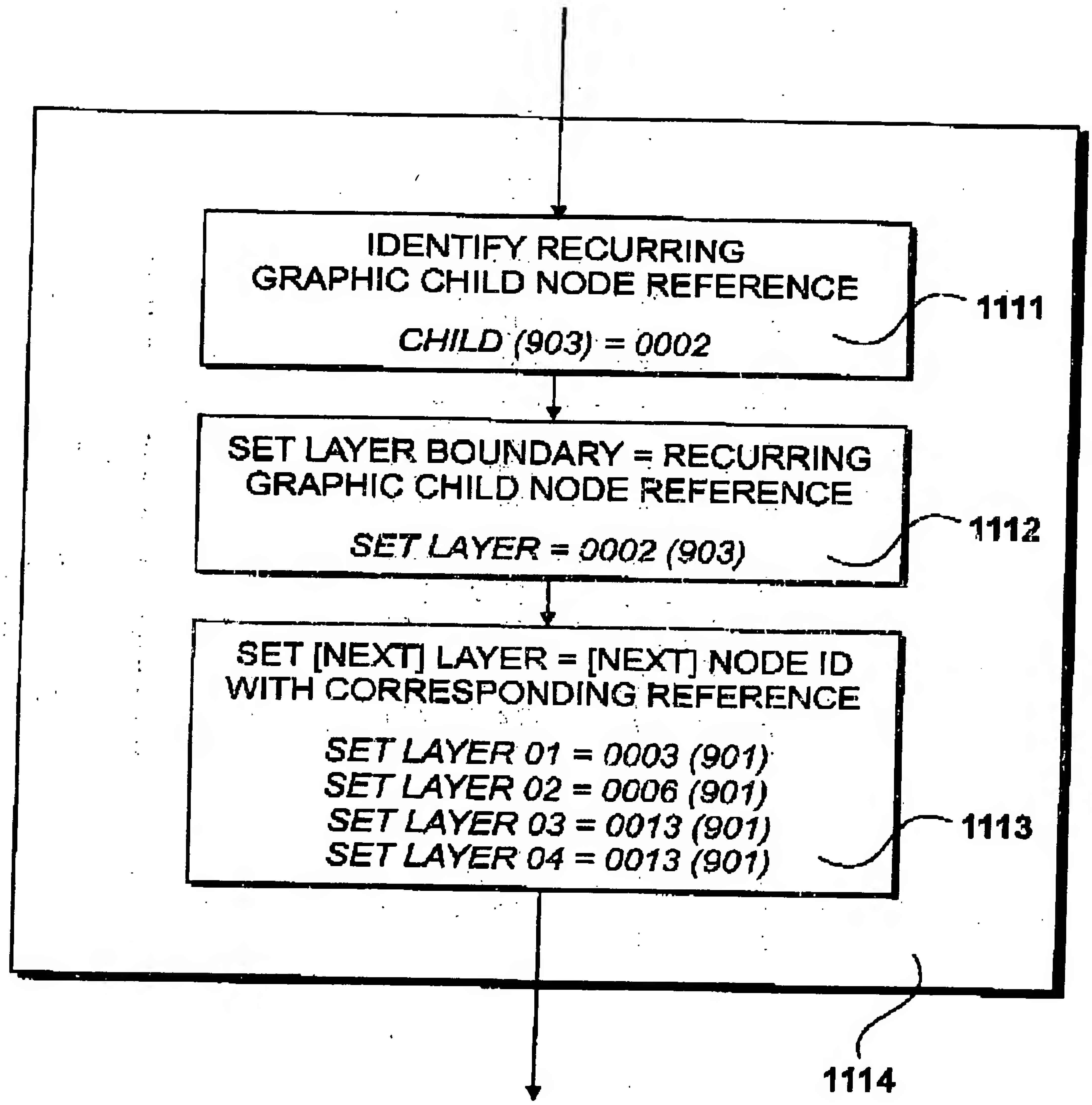

IDENTIFY RECURRING
GRAPHIC CHILD NODE REFERENCE
CHILD (903) = 0002
1111
SET LAYER BOUNDARY = RECURRING
GRAPHIC CHILD NODE REFERENCE
SET LAYER = 0002 (903)
1112
SET [NEXT] LAYER = [NEXT] NODE ID
WITH CORRESPONDING REFERENCE
SET LAYER 01 = 0003 (901)
SET LAYER 02 = 0006 (901)
SET LAYER 03 = 0013 (901)
SET LAYER 04 = 0013 (901)
1113
1114

Figure 11b

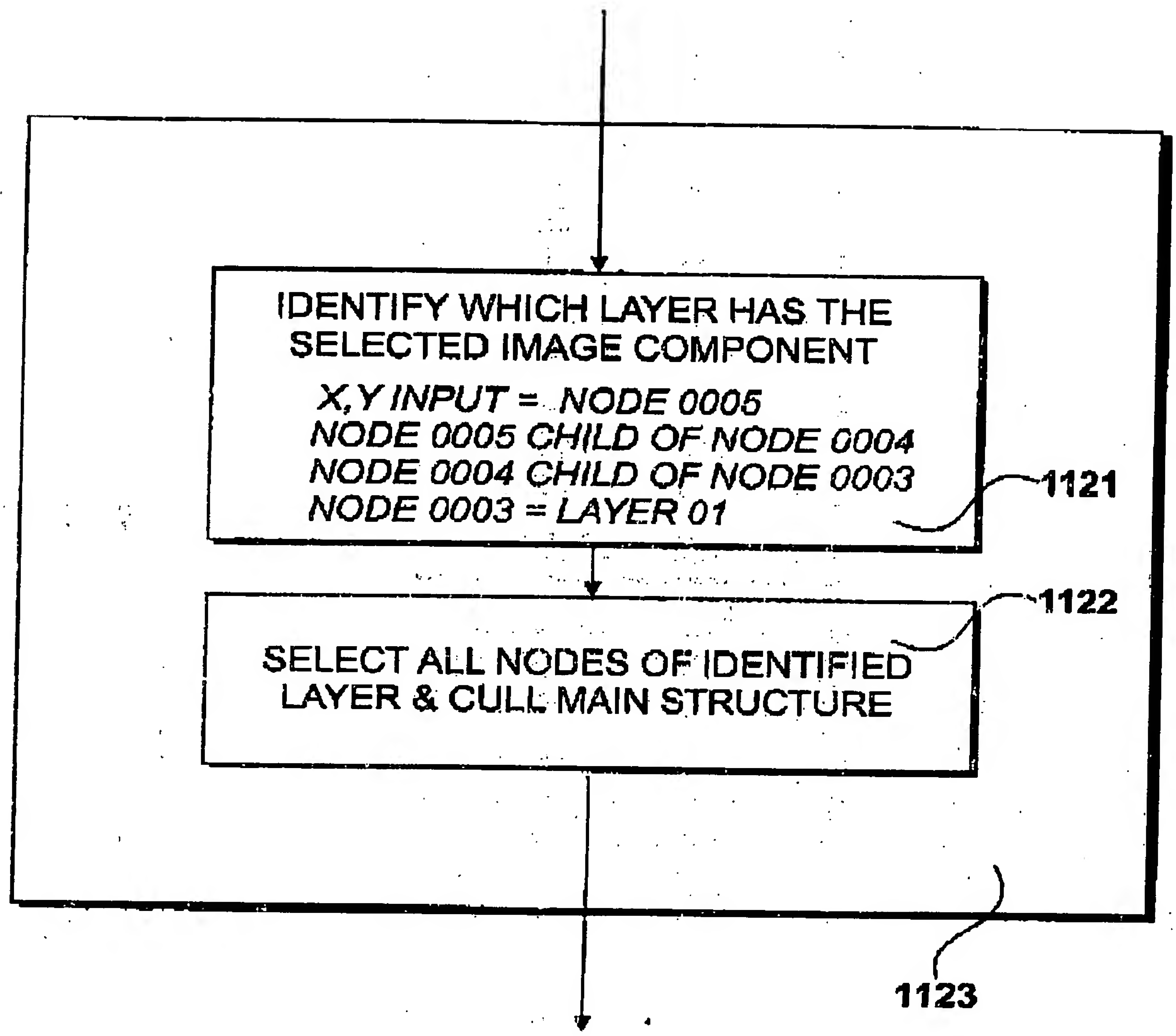

IDENTIFY WHICH LAYER HAS THE SELECTED IMAGE COMPONENT
X,Y INPUT = NODE 0005
NODE 0005 CHILD OF NODE 0004
NODE 0004 CHILD OF NODE 0003
NODE 0003 = LAYER 01
1121
1122
SELECT ALL NODES OF IDENTIFIED LAYER & CULL MAIN STRUCTURE
1123

Figure 11c

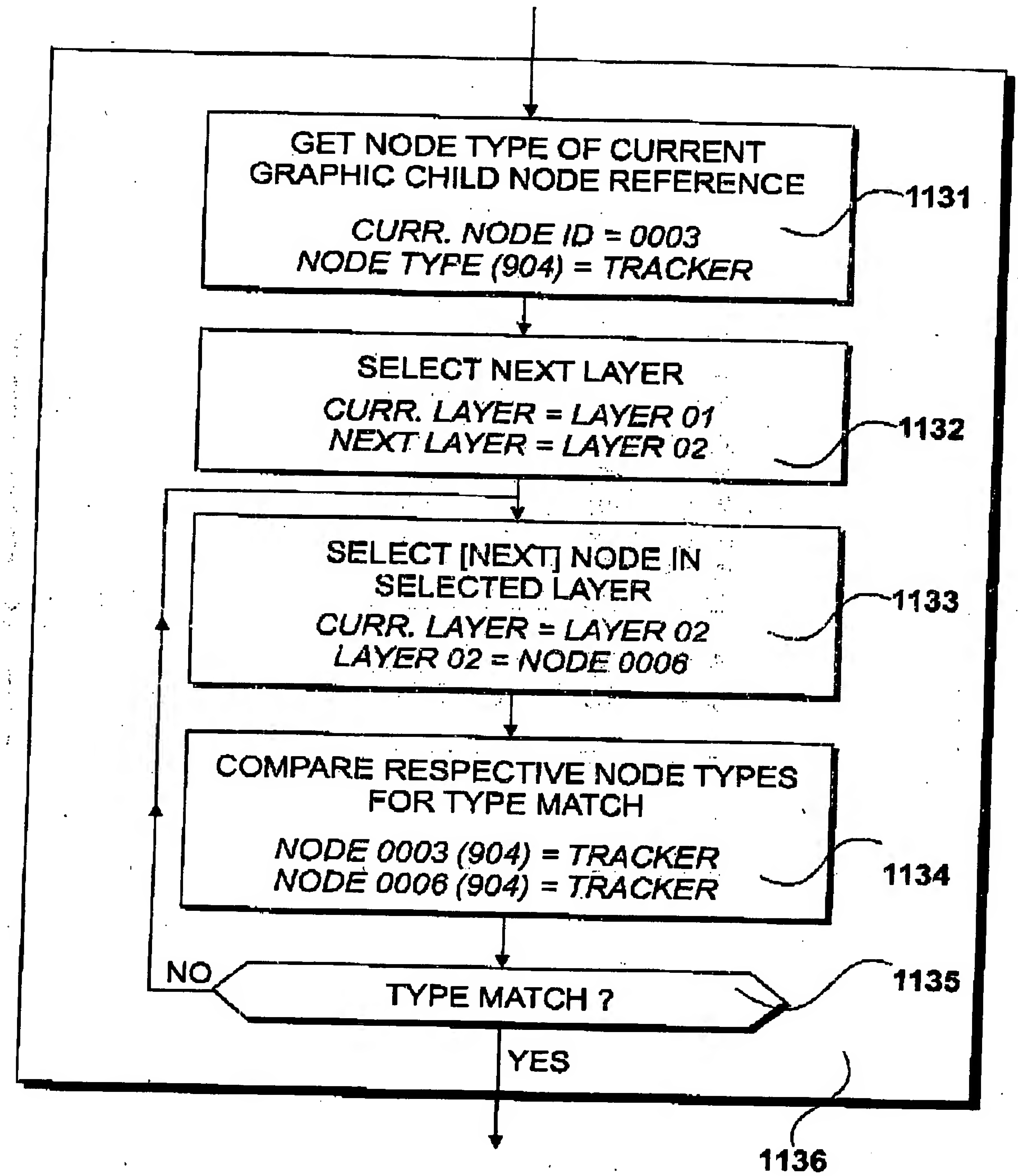

GET NODE TYPE OF CURRENT GRAPHIC CHILD NODE REFERENCE
CURR. NODE ID = 0003
NODE TYPE (904) = TRACKER
1131
SELECT NEXT LAYER
CURR. LAYER = LAYER 01
NEXT LAYER = LAYER 02
1132
SELECT [NEXT] NODE IN SELECTED LAYER
CURR. LAYER = LAYER 02
LAYER 02 = NODE 0006
1133
COMPARE RESPECTIVE NODE TYPES FOR TYPE MATCH
NODE 0003 (904) = TRACKER
NODE 0006 (904) = TRACKER
1134
NO
TYPE MATCH ?
1135
YES
1136

Figure 11d

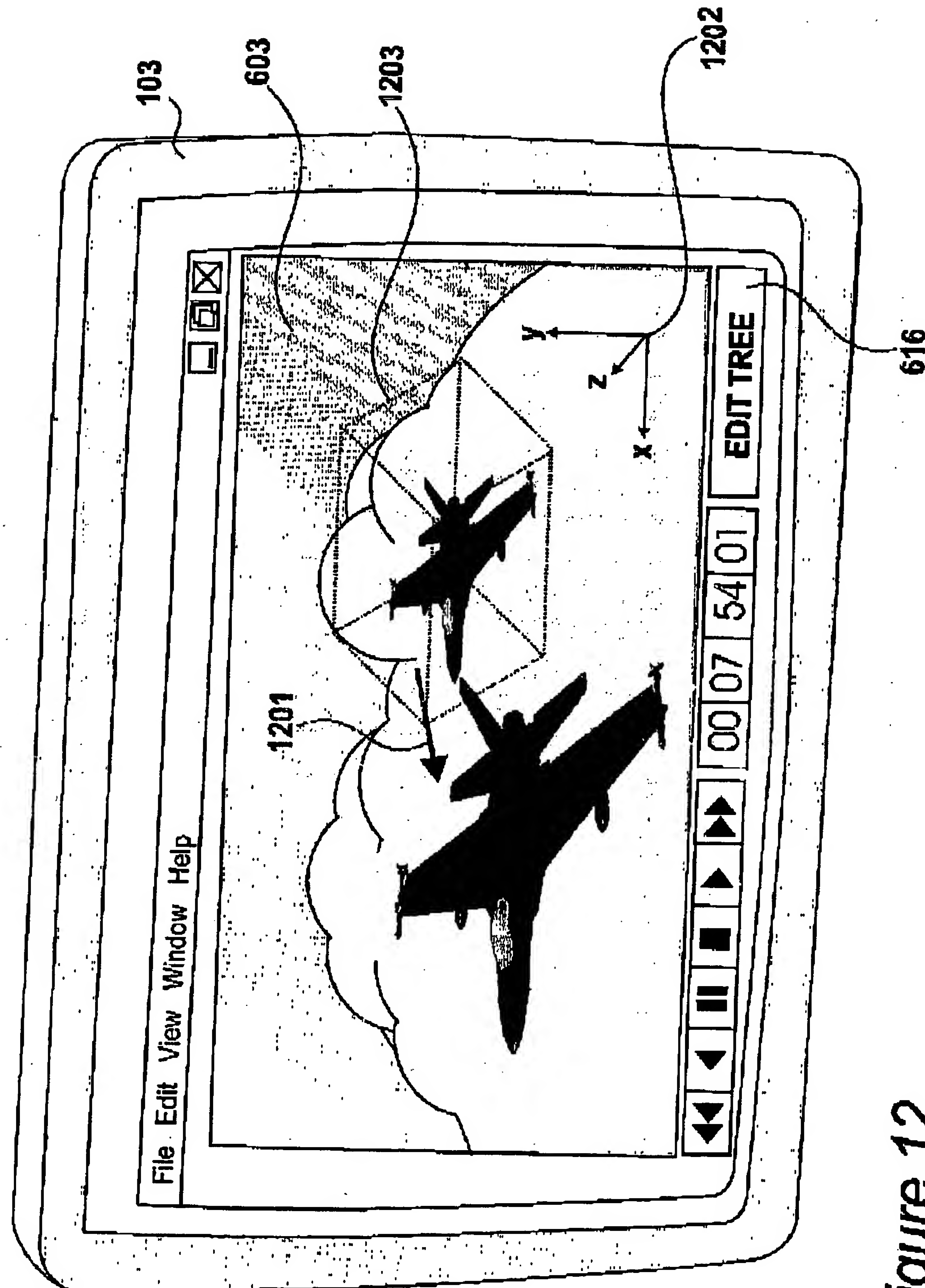

103
603
1203
1202
616
1201
File Edit View Window Help
y
z
x
EDIT TREE
00 07 54 01

Figure 12

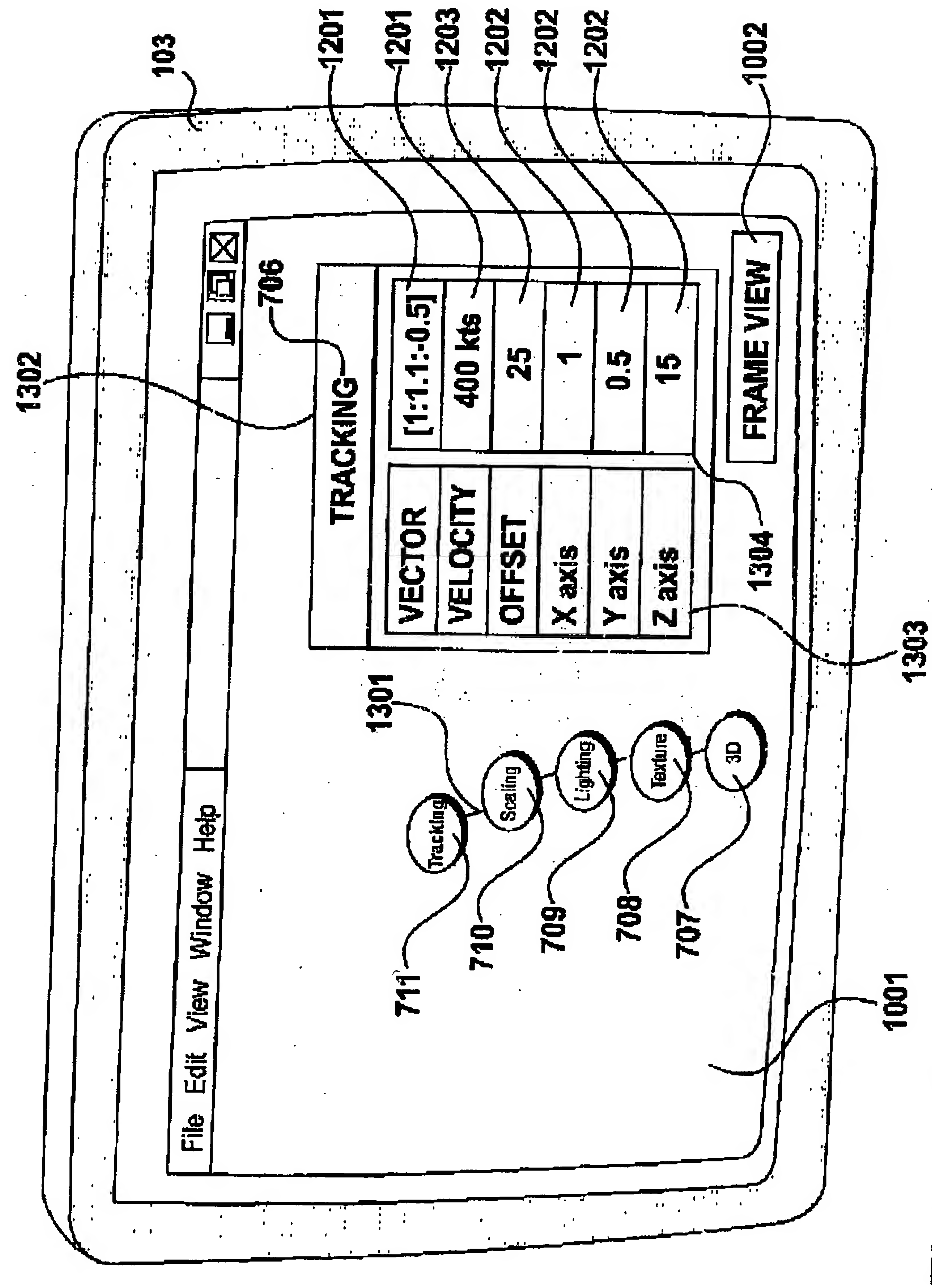

103
1302
706
TRACKING
VECTOR
VELOCITY
OFFSET
X axis
Y axis
Z axis
[1:1.1:-0.5]
400 kts
25
1
0.5
15
1201
1201
1203
1202
1202
1202
1002
FRAME VIEW
1304
1303
1301
File Edit View Window Help
Tracking
Scaling
Lighting
Texture
3D
711
710
709
708
707
1001

Figure 13

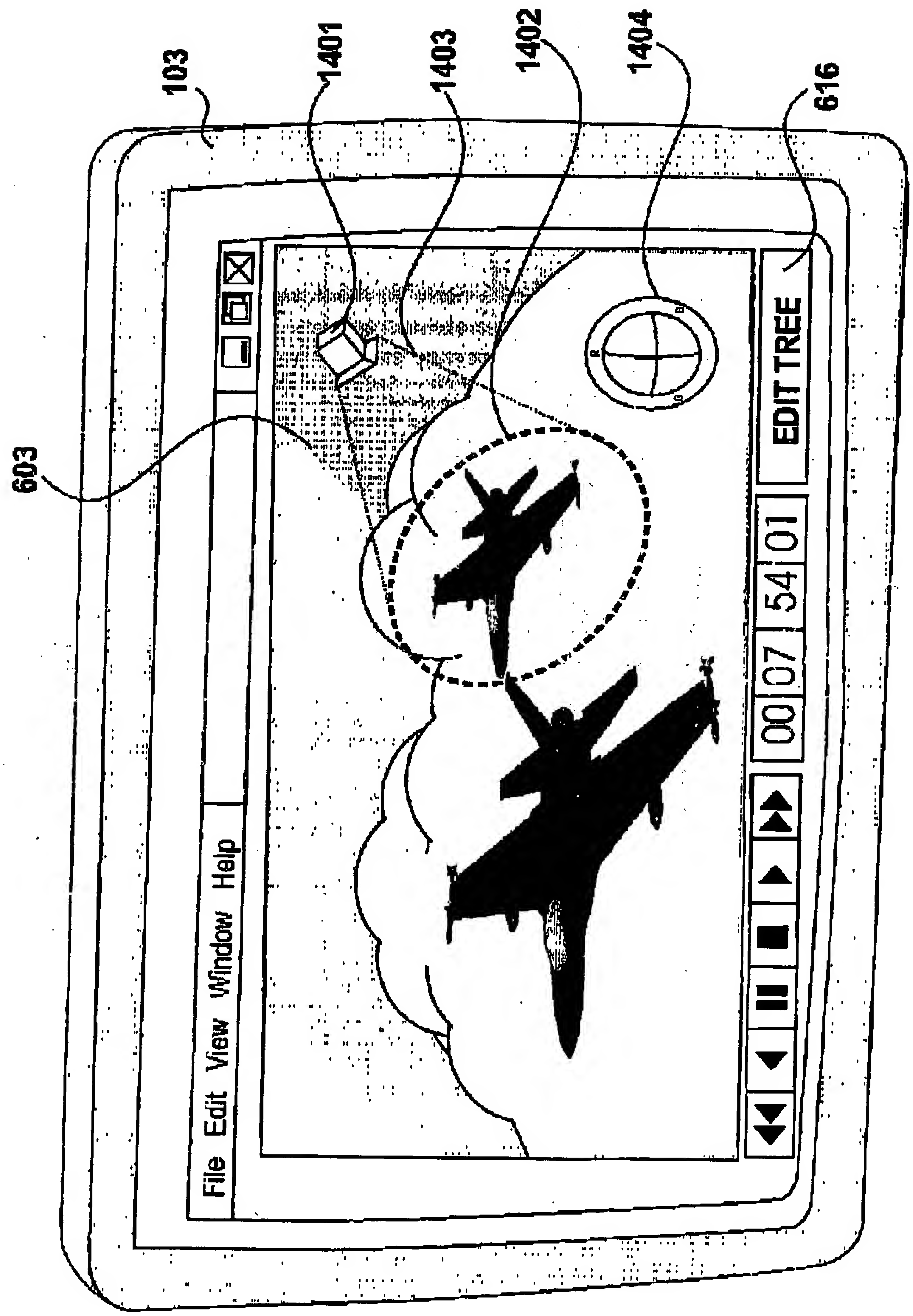

103
1401
1403
1402
1404
616
603
File Edit View Window Help
EDIT TREE
00 07 54 01

Figure 14

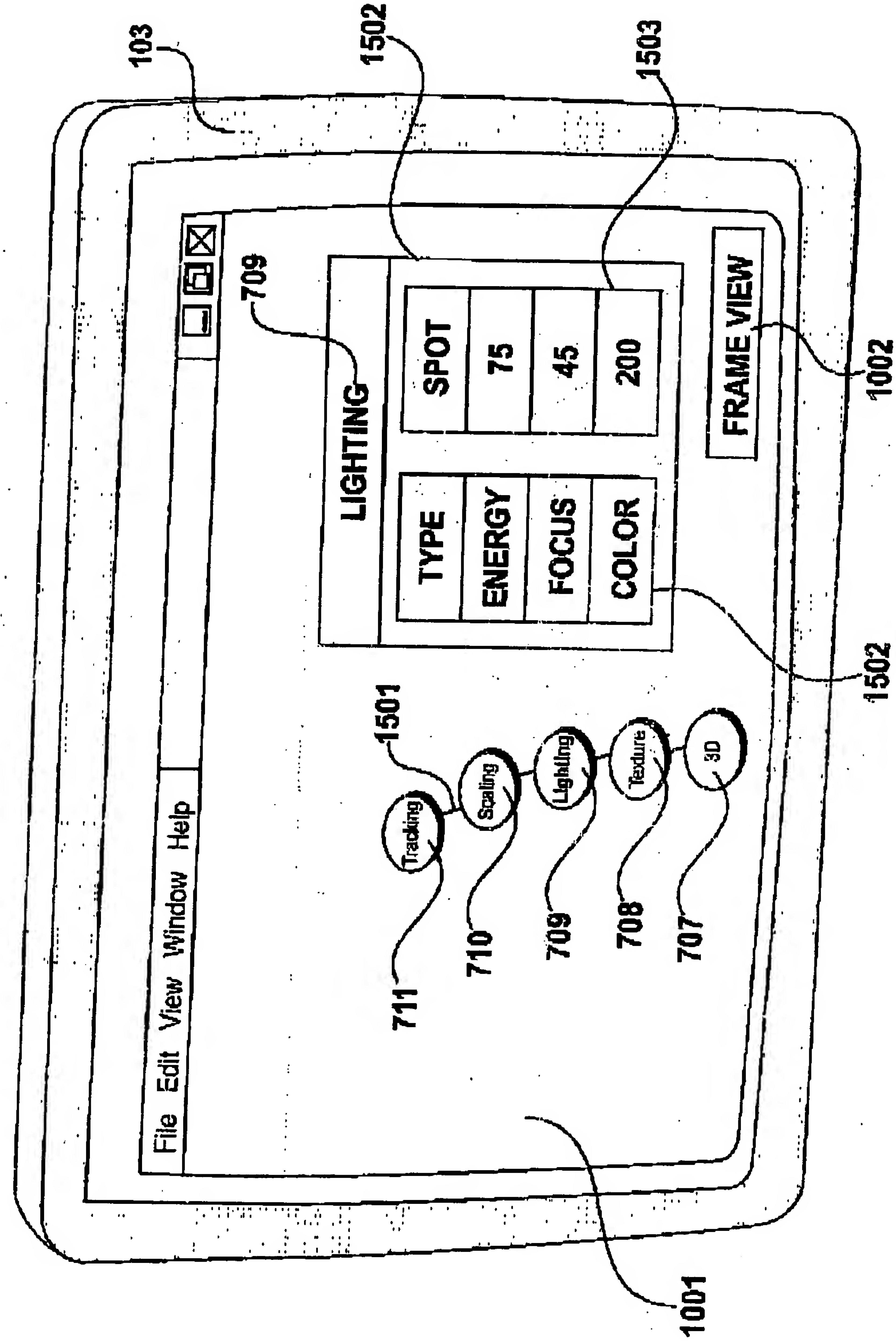

103
1502
1503
709
LIGHTING
TYPE
ENERGY
FOCUS
COLOR
SPOT
75
45
200
FRAME VIEW
1002
1502
1501
Tracking
Scaling
Lighting
Texture
3D
711
710
709
708
707
File Edit View Window Help
1001

Figure 15

2034-P578-GB

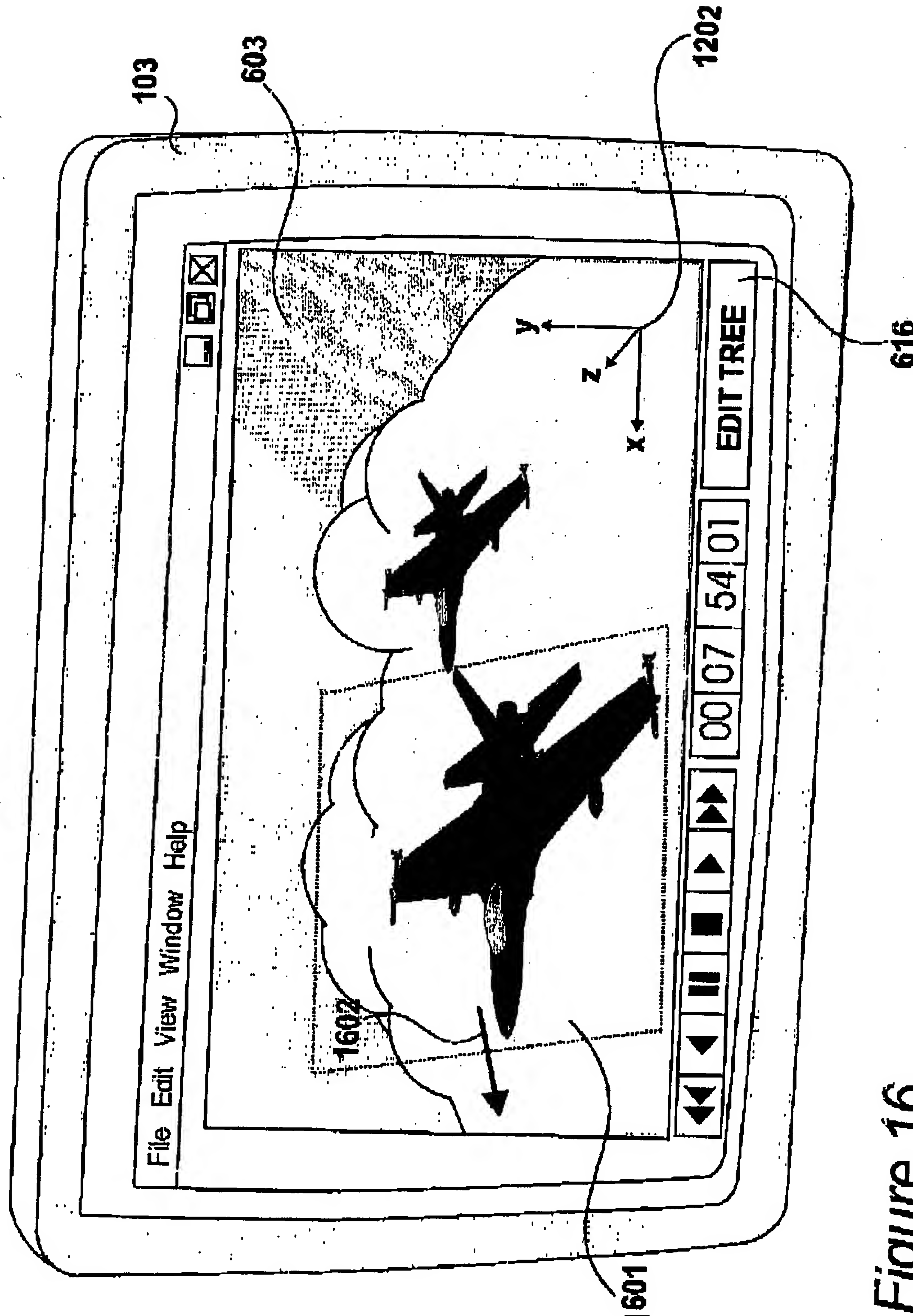


Figure 16

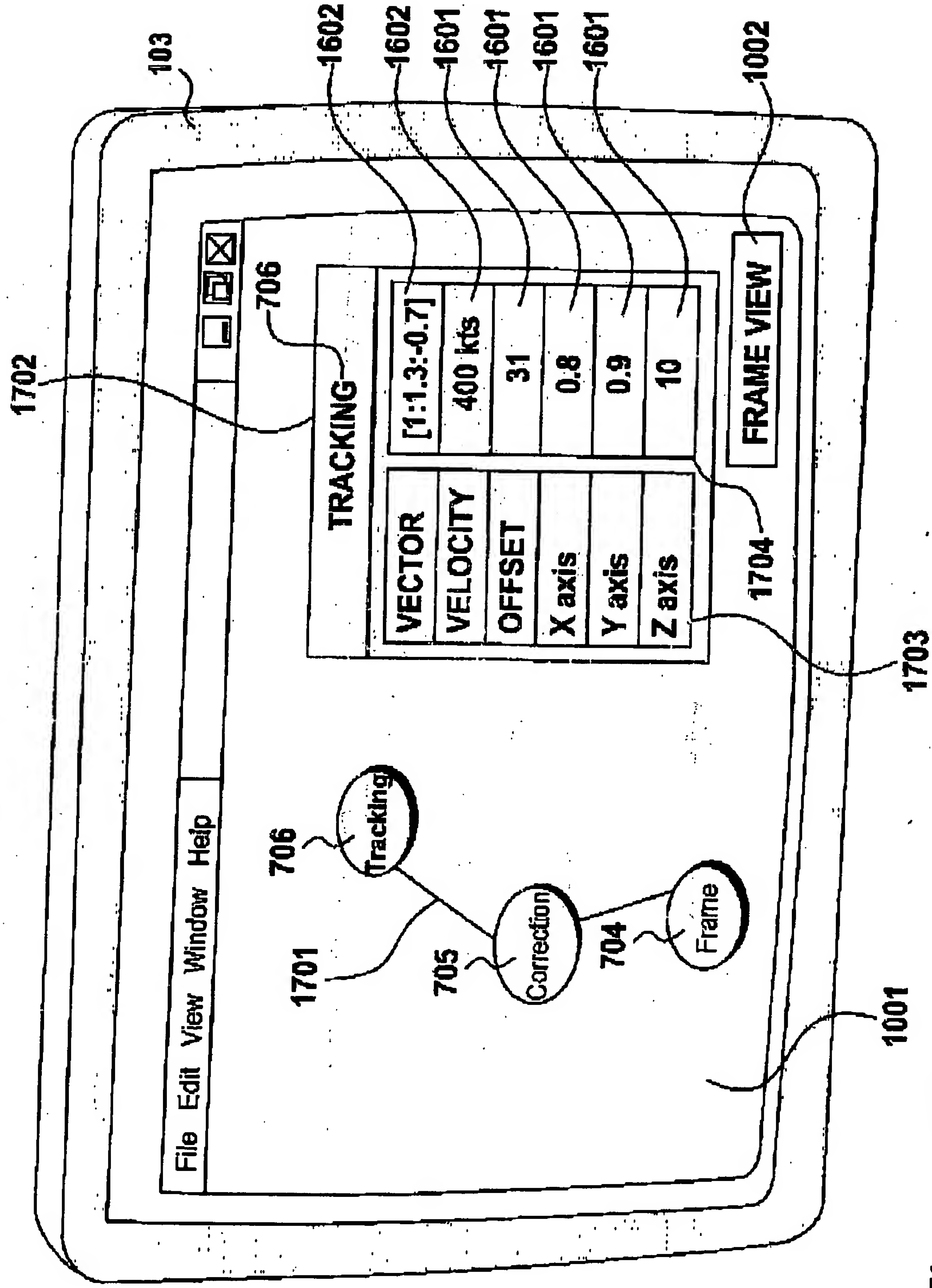

103
1602
1602
1601
1601
1601
1601
1002
1702
706
TRACKING
VECTOR
VELOCITY
OFFSET
X axis
Y axis
Z axis
[1:1.3:-0.7]
400 kts
31
0.8
0.9
10
FRAME VIEW
1704
1703
File Edit View Window Help
706
Tracking
1701
705
Correction
704
Frame
1001

Figure 17